

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam melaksanakan praktek kerja magang dalam *working unit* Financial Solution Delivery, dilakukan koordinasi serta komunikasi dalam bentuk tim yang terdiri dari 25 orang yang merupakan *software tester*, serta 13 orang yang merupakan *developer* untuk proyek perusahaan Askrindo dalam pengembangan *Askrindo Core System*. Tugas yang diberikan sebagai *developer* adalah pembuatan tampilan tatap muka (*User Interface*) *Askrindo Core System*, alur logika program, dan pembuatan database sesuai bisnis proses yang diterjemahkan sebagai *data flow diagram* dan flowchart oleh *system analyst*. Kemudian untuk tugas yang diberikan sebagai *software tester* adalah melakukan pengujian kelayakan atas *Askrindo Core System* yang dikembangkan dalam segi tatap muka, analisa data *input-output* baik secara lokal atau antar *API (Application Program Interface)*, analisa terhadap basis data (*database*) yang telah dibuat oleh *developer* terhadap bisnis proses yang telah disampaikan berdasarkan *business analyst* sehingga mengurangi adanya malfungsi atau *defect* ketika sistem tersebut digunakan oleh perusahaan Askrindo dalam *server production*. Untuk tim software tester, dibawah oleh bapak Zein Zenith yang merupakan kepala dari bagian software tester sekaligus pembimbing lapangan praktek kerja magang.

3.2 Tugas yang Dilakukan

Dalam pelaksanaan praktek kerja magang dalam *working unit* Financial Solution Delivery, terdapat tugas yang diberikan oleh kepala *software tester* sebagai berikut:

1. Mempelajari Bisnis proses perusahaan Askrindo terkait dengan asuransi yang dibagi menjadi 4 modul besar dalam *Askrindo Core System*, yaitu modul Marketing, modul *Underwriting*, modul Reasuransi, dan modul Klaim.
2. Mempelajari struktur database *SQL Server* dari *Askrindo Core System* dan query data yang diperlukan
3. Mempelajari *Jenkins* yang digunakan untuk melakukan *build* dan *deploy* sistem kedalam *environment* yang dibutuhkan.
4. Melakukan *testing* sesuai dengan test scenario serta dokumentasi sistem dalam *environment staging*, UAT, migrasi apabila terjadi malfungsi, *bug* atau defect.
5. Melakukan *reproduce bug* apabila dalam *environment production* terjadi dan ditemukan oleh pengguna *Askrindo Core System* secara langsung dalam *environment* migrasi dan melakukan analisa dari segi bisnis proses dan teknis.

6. Melakukan pemecahan masalah (*problem solving*) serta saran untuk developer baik dalam alur sistem, pengembangan *script* source code serta *script* query.

Dalam pelaksanaan praktek kerja magang sebagai software tester, dilakukan metode pendekatan *white box testing* yang berarti melakukan pengujian dengan mengamati struktur serta alur bagaimana data masuk serta keluarnya pada sistem tersebut. Karena dilakukan metode pendekatan *white box testing*, *software tester* serta *developer* dituntut untuk saling berkomunikasi sehingga pengembangan dapat berjalan dengan cepat.

3.3 Uraian Pelaksanaan Magang

Untuk menyelesaikan tugas yang diberikan, terdapat kegiatan-kegiatan yang menunjang untuk tugas tersebut. Setelah menjabarkan tugas yang dilakukan pada sub-bab sebelumnya, terdapat kegiatan yang dilakukan setiap minggunya.

3.3.1 Minggu Satu, Pembelajaran Bisnis Proses *Client*, ACS System, Test Scenario, dan Database Server

Kegiatan yang dilakukan pada minggu pertama ke dalam tim adalah pengenalan akan pelanggan atau biasa disebut *client* bersama bapak Zein Zenith yang merupakan kepala *testing* dari tim *software tester*. Perusahaan Askrindo merupakan perusahaan Badan Usaha Milik Negara (BUMN) yang menyediakan layanan asuransi.

Dalam kegiatan pengenalan bisnis proses ini, terdapat beberapa istilah-istilah inti yang harus dimengerti terlebih dahulu. Berikut istilah-istilah inti yang dijelaskan berdasarkan kepala software tester:

1. Asuransi, yang merupakan sebuah bisnis yang merupakan perlindungan secara finansial atau penggantian kerugian secara finansial untuk objek yang ditanggungkan seperti jiwa, properti, kesehatan dan sebagainya atas kejadian yang tidak dapat diduga seperti, kematian, kehilangan, atau sakit, bencana alam dan sebagainya yang diharuskan membayar premi dalam waktu yang ditentukan
2. *Underwriting*, yang merupakan sebuah proses menganalisa, dan penyaringan risiko yang dimiliki oleh calon pelanggan atau nasabah asuransi.
3. Akseptasi, yang merupakan proses penerimaan resiko yang dimiliki oleh pelanggan atau nasabah asuransi.
4. Polis, yang merupakan sebuah perjanjian atau kesepakatan asuransi atau pertanggungan untuk kedua belah pihak yang dimana disini merupakan kesepakatan atas perusahaan asuransi dengan nasabah asuransi yang berisi ketentuan yang bersifat tertulis.
5. Premi, yang merupakan sejumlah uang yang harus dibayarkan kepada perusahaan asuransi sebagai kewajiban nasabah atas keikutsertaannya

dalam asuransi. Besaran Premi tersebut dilihat berdasarkan keadaan dari tertanggung yang merupakan nasabah sendiri.

6. Deductible, merupakan resiko yang harus ditanggung sendiri oleh tertanggung.
7. Warranty, yang merupakan kondisi yang dapat membatalkan kontrak atau polis tersebut apabila kondisi tersebut terjadi. Contoh tidak ada minyak solar atau bensin di dalam gudang yang tertulis, maka apabila tertanggung mengalami kerugian dan terdapat pada kondisi seperti demikian, maka ketentuan untuk melakukan klaim yang tertulis pada polis tersebut tidak akan disetujui
8. Klausula, merupakan batas tanggung jawab penanggung sewaktu membayarkan kerusakan apabila terjadi peristiwa yang mengakibatkan kerugian tersebut yang perjanjiannya tertulis di dalam polis secara tegas dan jelas. Klausula tersebut dibagi menjadi beberapa bagian antara lain:
 - a. Klausula Primer Risque, merupakan klausula yang menyatakan jika nilai asuransi atau nominal yang diasuransikan lebih kecil daripada nilai atau harga awal benda tersebut, maka penanggung akan membayar ganti kerugian seluruhnya hingga nilai maksimum jumlah yang diasuransikan, sebagai contoh apabila perusahaan A melakukan asuransi aset kendaraan bernilai 2 miliar Rupiah dengan pertanggungan senilai 500 juta Rupiah, maka apabila ada kerugian sebesar seluruh aset kendaraan tersebut, maka penanggung hanya

membayar senilai 500 juta Rupiah. Dasar aturan ini berada pada Kitab Undang-Undang Hukum Dagang pasal 253, ayat 3.

- b. Klausula All Risk, merupakan klausula yang menyatakan penanggung akan menanggung secara seluruhnya terhadap benda yang diasuransikan, walaupun kerugian yang terjadi dibawah nilai objek yang diasuransikan. Contoh, perusahaan A melakukan asuransi aset kendaraan senilai 2 miliar Rupiah dengan objek pertanggungan ketentuan klausula All Risk, maka apabila ada kejadian yang menimpa yang membuat kerugian sebesar 100 juta rupiah, maka penanggung tetap mengganti sebesar 2 miliar Rupiah untuk diberikan kepada tertanggung. Dasar aturan ini berada pada Kitab Undang-Undang Hukum Dagang Pasal 249.
- c. Klausula All Seen (sudah diketahui), merupakan klausula yang menyatakan bahwa penanggung telah mengetahui kondisi, keadaan, konstruksi, letak, dan cara pengaturan atau pemakaian bangunan yang diasuransikan, dimana klausula ini diperuntukan pada asuransi kebakaran.
- d. Klausula Renunsiasi, merupakan klausula yang menyatakan penanggung tidak bisa menggugat yang tertanggung yaitu pihak ketiga yang menyebabkan tertanggung mengalami kerugian sesuai KUHD Pasal 251. Apabila terjadi kejadian yang tidak terduga atau tidak pasti yang menyebabkan kerugian, tertanggung

dapat tidak memberitahu keadaan objek yang diasuransikan kepada penanggung. Namun apabila hakim memutuskan harus diberlakukan secara jujur atau itikad baik dan dengan kebiasaan, maka penanggung dapat menuntut pihak ketiga yang membuat kerugian pada tertanggung.

- e. Reasuransi, yang berarti asuransi dari nasabah yang tertanggung kembali diasuransikan kepada perusahaan asuransi pihak ketiga lain oleh penanggung dari asuransi nasabah tersebut atau melakukan asuransi ulang terhadap asuransi dari tertanggung yang dilakukan oleh penanggung atau perusahaan kepada pihak penanggung ketiga.
- f. Koasuransi, yang berarti pertanggungan asuransi terhadap objek yang diasuransikan, resiko tersebut dibagi dengan tertanggung apabila terjadi suatu kejadian yang tidak terduga terhadap tertanggung yang menyebabkan kerugian, maka tertanggung dan penanggung akan sama-sama membayar atas kerugian tersebut
- g. Aktuaria, yang merupakan ilmu statistik, ekonomi, keuangan, serta matematika yang dikombinasi menjadi salah satu cabang ilmu matematika yang berguna untuk memprediksi pengelolaan risiko keuangan mendatang.

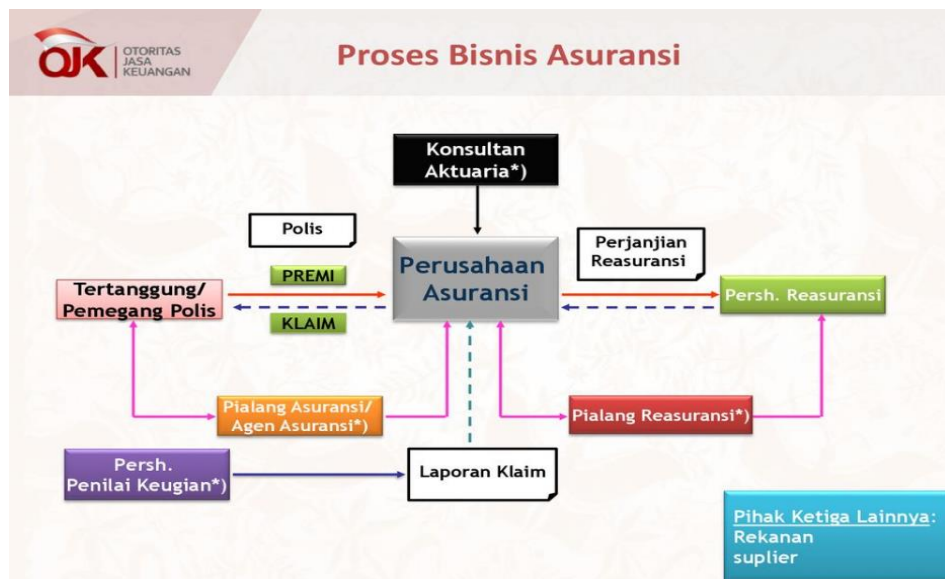
Setelah kepala *software tester* menjelaskan istilah inti yang telah dijabarkan sebelumnya, maka dijelaskan juga bisnis proses perusahaan Askrindo yang terkait

dalam Asuransi ini. Proses bisnis tersebut secara umum memiliki alur seperti berikut:

- a. Terdapat pihak-pihak yang terlibat dalam bisnis proses ini yaitu pihak perusahaan Askrindo sendiri, serta pihak luar seperti Tertanggung atau nasabah, perusahaan penilai kerugian, konsultan aktuaria, perusahaan reasuransi, supplier, agen asuransi, agen reasuransi. Kemudian terdapat dokumen yang berupa polis, laporan kalim, dan perjanjian asuransi
- b. Pertama dimulai dari nasabah yang telah mendaftarkan dalam layanan asuransi perusahaan Askrindo yang mengajukan pertanggungan terhadap objek yang diasuransikan kepada perusahaan Askrindo dengan agen asuransi atau tanpa pihak perantara dengan mengajukan sendiri ke perusahaan Askrindo di kantor cabang terdekat.
- c. Lalu konsultan aktuaria dalam perusahaan Askrindo akan melakukan analisa resiko terhadap objek yang akan ditanggungkan kepada perusahaan Askrindo sebelum permintaan pertanggungan nasabah disetujui yang nantinya perusahaan Askrindo menerbitkan polis yang berisi klausula klausula serta aturan perjanjian lainnya yang disepakati untuk objek yang diasuransikan kepada perusahaan Askrindo dimana yang mengharuskan nasabah membayar premi atau iuran asuransi kepada perusahaan Askrindo.

- d. Perusahaan Askrindo tentunya mempunyai batasan spesifik untuk setiap objek yang diasuransikan, oleh karena itu perusahaan perlu pihak penanggung ketiga yaitu perusahaan reasuransi lainnya. Untuk perjanjian reasuransi dengan pihak penanggung ketiga atau perusahaan reasuransi, diperlukan kesepakatan dengan perusahaan reasuransi tersebut yang terkait dengan ketentuan, *treaty* yang ada didalamnya dan ketentuan-ketentuan lainnya yang terbentuk dalam dokumen perjanjian reasuransi. Perusahaan Askrindo dapat berkomunikasi terkait perjanjian reasuransi terhadap perusahaan reasuransi dengan agen atau perantara dari perusahaan reasuransi tersebut atau bisa juga secara langsung berkomunikasi dengan perusahaan reasuransi tersebut.
- e. Setelah polis tersebut jadi dan diberikan kepada tertanggung atau nasabah oleh perusahaan Askrindo, maka apabila nasabah melakukan klaim atas objek yang diasuransikan, maka tertanggung dapat langsung menghubungi perusahaan Askrindo atau dengan agen asuransi yang menghubungkan perusahaan Askrindo.
- f. Ketika tertanggung mengajukan klaim ke perusahaan Askrindo, maka klaim tersebut akan dilakukan analisa atau *review* oleh pihak perusahaan Askrindo sendiri bagian klaim serta perusahaan penilai kerugian pihak ketiga untuk disetujui atau tidak atas klaim tertanggung. Dalam proses ini akan direview berdasarkan polis

yang ada, serta ketentuan-ketentuan yang ada dari dokumen perjanjian reasuransi, dan hasil analisa yang terkumpul di dalam dokumen laporan klaim.



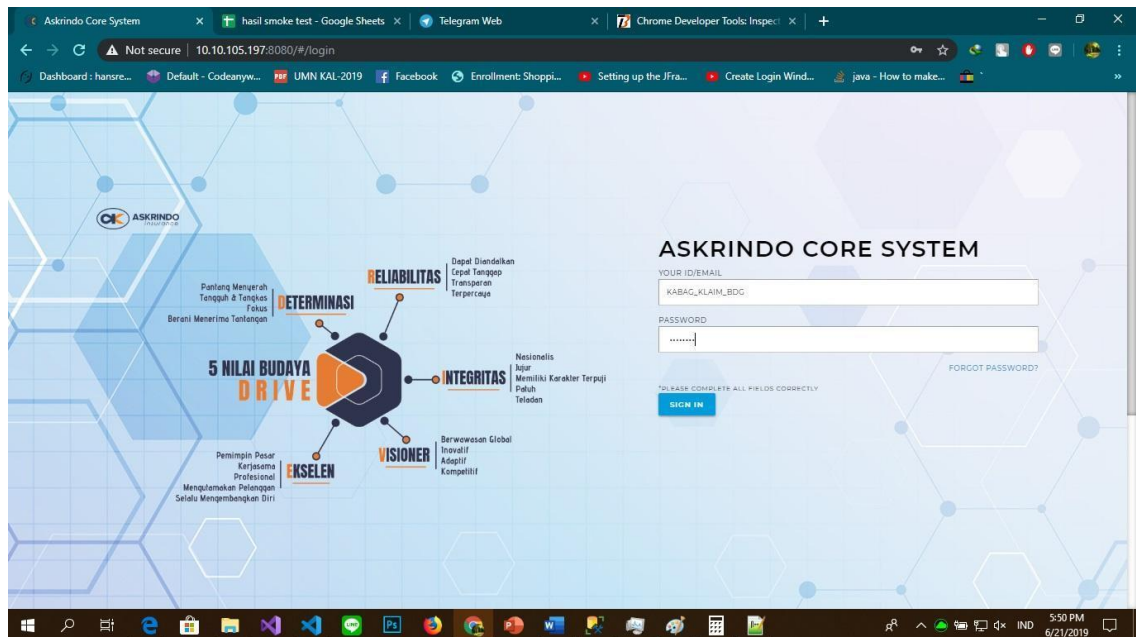
Gambar 3.1. Diagram bisnis proses perusahaan Askrindo

Setelah dilakukan penjelasan terkait dengan bisnis proses dan istilah inti yang perlu diketahui dan dimengerti, maka kepala *software tester* menjelaskan sistem yang sedang dikembangkan dan untuk dipelajari, yaitu *Askrindo Core System* (ACS) dan *Askrindo Financial Management System* (FMS). *Askrindo Core System* Merupakan Sistem yang berfungsi untuk mengumpulkan data, mengolah data terkait dengan data asuransi yang memiliki keluaran seperti polis, laporan klaim, Laporan Kerugian Sementara (LKS), LPK (Laporan Persetujuan Kerugian). *Financial Management System* merupakan sistem yang berfungsi dalam mengatur data keuangan, serta manajemen laporan yang terkait dengan keuangan perusahaan

Askrindo dalam bisnis asuransi seperti pengaturan nilai mata uang asing, nilai atau nominal dasar premi, dan pembukuan atau penjurnalan, serta manajemen risiko terhadap nominal terhadap objek yang diasuransikan yang terintegrasi dengan sistem ACS. Sistem ini berbasis *website* yang dapat diakses dari berbagai kantor cabang perusahaan Askrindo karena sistem tersebut memiliki *server* yang secara sentral atau terpusat pada Kemayoran, Jakarta Pusat. Sesuai tugas yang diberikan oleh kepala *software tester*, maka akan dipelajari untuk sistem ACS saja. Sistem ACS terbagi menjadi 4 modul besar yaitu:

1. Modul Marketing
2. Modul Akseptasi atau *Underwriting*
3. Modul Reasuransi
4. Modul Klaim

Pengguna sistem ACS ini merupakan pekerja perusahaan Askrindo yang dikelompokkan berdasarkan peran atau hak akses yang telah ditetapkan oleh pelanggan Askrindo serta kedudukan divisi kepada karyawan sesuai yang mengoperasikan sistem tersebut. Untuk mengoperasikan sistem ACS, maka diperlukan akun untuk pengguna sistem tersebut serta berguna juga untuk membatasi hak akses dari pengguna sistem ACS.



Gambar 3.2. Tampilan halaman *web* sistem ACS untuk *login*

Pada gambar 3.2 merupakan contoh halaman *web* yang digunakan oleh pengguna sistem untuk *login* berdasarkan kredensial yang telah diberikan oleh perusahaan Askrindo.

3.3.1.1 Modul *Marketing*

Modul *Marketing* merupakan modul yang memiliki tujuan untuk pengelolaan atau pengaturan data yang terkait dengan master data baik nasabah maupun semua pihak yang melakukan hubungan bisnis dengan perusahaan Askrindo. Menurut kepala *software tester*, dalam pengelompokan data yang terkait dengan partner bisnis, dilakukan pengelompokan seperti berikut:

- a. Perseorangan

- b. Korporasi BUMN, BUMD
- c. Korporasi Swasta
- d. Bank
- e. Leasing
- f. Agen
- g. Broker
- h. Asuransi
- i. Reasuransi

Sesuai dengan ketentuan perusahaan Askrindo, modul *Marketing* mempunyai fitur diantaranya adalah:

- a. Fitur Bisnis Partner yang berfungsi untuk melakukan pencatatan data nasabah perusahaan Askrindo baik dalam perubahan nasabah baru atau penambahan nasabah baru, Data profile nasabah, Data histori transaksi, Fasilitas *attachment* dokumen dari nasabah terkait informasi diri.
- b. Fitur Blacklist yang berfungsi untuk melakukan pemblokiran partner bisnis atau nasabah agar mencegah dalam menggunakan layanan asuransi Askrindo atau bertransaksi dengan perusahaan Askrindo.



Gambar 3.3. Informasi yang diperlukan dalam fitur *Business Partner*

Berdasarkan kepala *software tester*, dalam pencatatan pihak luar sebagai partner bisnis, tentunya terdapat kesepakatan antara dua belah pihak baik dari perusahaan Askrindo maupun pihak luar yang disebut dengan perjanjian. Perjanjian antara perusahaan Askrindo dengan partner bisnis, dapat dikategorikan menjadi tiga jenis perjanjian, antara lain:

- a. Perjanjian reasuransi, yang merupakan perjanjian dengan pihak luar yang memberikan atau menyediakan bisnis reasuransi
- b. Perjanjian broker atau agen, dimana perjanjian tersebut berlaku antara pihak perusahaan Askrindo dengan pihak ketiga yang menghubungkan perusahaan dengan nasabah sebenarnya.
- c. Perjanjian pihak utama, di mana perjanjian tersebut berlaku antara pihak perusahaan Askrindo dengan pihak bertanggung secara langsung seperti bank, perseorangan, *leasing*, dan lain-lain.

Data yang telah tersimpan melalui modul *Marketing*, dapat digunakan sebagai informasi dasar dari tiga modul lainnya yang berperan menjadi *input* atau

masukan. Dalam pembuatan polis yang merupakan bagian proses dalam modul *Underwriting* yang akan dibahas nanti, membutuhkan informasi terkait partner bisnis yang telah diolah dan tercatat pada modul *Marketing*. Oleh karena itu, modul *Marketing* merupakan tahap awal ketika pihak luar ingin membangun hubungan bisnis dengan perusahaan Askrindo.

3.3.1.2 Modul *Underwriting*

Modul akseptasi atau disebut juga modul *underwriting* berfungsi untuk melakukan pencatatan SPPA yang merupakan singkatan dari Surat Permohonan Penutupan Asuransi kemudian untuk dianalisa untuk disetujui yang sebagai dasar penerbitan polis dan fungsi administrasi lainnya yang dibutuhkan terkait dengan pencetakan dokumen terkait polis. SPPA merupakan surat permohonan dari nasabah asuransi atau tertanggung untuk mendaftarkan objek yang ditanggungkan yang merupakan dokumen dasar yang nantinya akan dilakukan analisa untuk pembuatan polis asuransi. SPPA diperuntukan untuk nasabah asuransi yang telah terdaftar pada dalam *Business Partner* yang terdapat pada dalam modul marketing yang telah dibahas sebelumnya. SPPA sendiri bisa dikatakan polis yang belum diakseptasi.

Dalam sistem ini, pengguna sistem ACS yang menerima data permohonan SPPA dari nasabah pelanggan, dapat melihat data profil calon tertanggung yang telah terdaftar melalui hasil proses pada modul marketing melakukan entry data awal yang dibutuhkan untuk membuat SPPA. Dalam pembuatan dokumen SPPA tentunya memiliki struktur yang terdiri dari:

- a. Tipe produk TOC (Terms and Condition)
- b. Keterangan Tertanggung
- c. Keterangan Pertanggungan
- d. Keterangan objek pertanggungan
- e. Premi dan pembayaran
- f. Klausula serta *warranty*

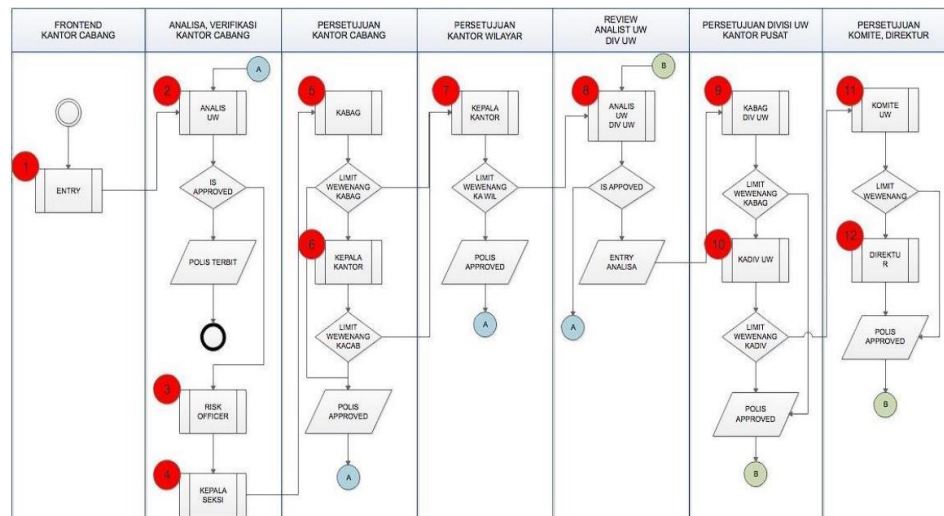
Kemudian modul *underwriting* membutuhkan pertukaran informasi dari antara modul reasuransi untuk mengambil data *treaty* atau *Spreading of Risk* (SOR) serta sistem FMS untuk mengambil data yang terkait dari nilai premi atau bobot premi ditambah dengan perhitungannya, perhitungan komisi, share koasuransi.

Dalam ketentuan perusahaan Askrindo, terdapat hak akses pengguna sistem ACS untuk melakukan approval SPPA yang tertunda berdasarkan hasil analisis cabang, atau analisis *Risk Officer* wilayah serta berdasarkan nominal objek pertanggungan tersebut. Berikut terdapat tabel yang menampilkan pengguna yang dapat melakukan persetujuan SPPA berdasarkan aturan perusahaan Askrindo.

Pengguna	Cabang Kelas 1	Cabang Kelas 2	Cabang Kelas 3	Cabang Kelas 4
KASIE	3.000.000.000	-	-	-
KABAG	5.000.000.000	4.500.000.000	4.000.000.000	3.500.000.000
PINCAB	7.000.000.000	6.500.000.000	6.000.000.000	5.500.000.000
PINWIL	10.000.000.000	9.500.000.000	9.000.000.000	8.500.000.000
	PUSAT			
KABAG PUSAT	13.000.000.000			
KADIV UDW	16.000.000.000			
KOMITE	19.000.000.000			
DIREKTUR	TIDAK ADA LIMIT			

Tabel 3.1 Tabel hak wewenang pengguna sistem ACS

Menurut kepala *software tester*, *workflow* atau alur kerja merupakan proses bisnis atau pengerjaan secara sistematis dimana informasi yang telah dibuat kemudian dikirimkan dari satu pihak lalu dikirimkan kembali kepada pihak yang lain untuk melakukan tindakan lanjutan berdasarkan prosedur yang berlaku dalam sebuah perusahaan atau organisasi. Dalam modul *Underwriting* pada sistem ACS, terdapat alur yang menjadi pedoman dalam pengembangan sistem berdasarkan ketentuan perusahaan Askrindo dalam langkah-langkah proses pembuatan polis yang berbentuk diagram *Swimlane*.



Gambar 3.4. Diagram *swimlane* terkait pembuatan polis beserta pihak yang terlibat

Dalam penerbitan polis terdapat proses yang perlu dilakukan dalam modul *underwriting*. Berikut proses dalam pembuatan polis berikut dengan penjelasannya:

- a. Proses *New*, merupakan proses pembuatan SPPA Baru, dimana terdapat sub-proses yaitu *create new*, *copy all*, *copy partial*. *Create new* merupakan proses pembuatan SPPA secara utuh dari awal atau menggunakan data baru secara seluruhnya. *Copy ALL* merupakan proses pembuatan SPPA dengan menggunakan data dari polis yang sudah terbit sebelumnya untuk menghemat proses pencatatan data nasabah. *Copy partial* adalah proses pembuatan SPPA secara mengambil sebagian data dari polis yang sudah ada untuk dimasukkan ke dalam SPPA baru dengan mengambil profil bertanggung data pemegang polis lama tanpa mengambil detail objek

pertanggungan pada polis sebelumnya untuk menghemat proses pencatatan data nasabah.

- b. Proses *Endorsement* Polis, merupakan proses perubahan data terhadap polis yang sudah jadi sebelumnya seperti penambahan objek pertanggungan, perubahan nama objek pertanggungan, atau nilai objek pertanggungan.
- c. Proses *Renewal* Polis, merupakan proses perpanjangan jangka waktu berlakunya sebuah polis yang sudah ada.

Proses paling penting yang harus dipahami dalam penerbitan polis adalah proses *New* yang merupakan akar dari proses-proses dalam pembuatan polis selanjutnya. Proses *New* dinyatakan akar dari proses-proses dalam pembuatan polis selanjutnya dikarenakan, proses seperti *Renewal Polis* serta *Endorsement Polis* membutuhkan *output* dari proses *New* tersebut yaitu merupakan polis yang sudah jadi.

Berdasarkan ketentuan perusahaan Askrindo, terdapat format penomoran dalam dokumen SPPA yang harus diikuti. Dalam nomor SPPA tersebut dapat mengandung informasi kode COB, tahun terbit polis, kode cabang penerbitan SPPA, sumber bisnis dari nasabah, nomor urut terbit SPPA, ditambah satu digit yang merupakan nomor pengecekan digit untuk sistem ACS.

	Urutan	Jumlah Digit	Keterangan
Kode COB	1	3	Alias (angka)
Tahun Terbit	2	2	(angka)
Kode Cabang	3	3	(angka)
Sumber bisnis	4	1	Agen/broker/direct bisnis(angka)
Nomor urut terbit	5	5	<i>Running number (incremental)</i>
Cek digit	6	1	Cek keaslian

Tabel 3.2. Tabel format penomoran SPPA

3.3.1.3 Modul Reasuransi

Modul Reasuransi atau *Reinsurance* merupakan modul yang berfungsi untuk mengatur atau mengelola data terkait dengan yang berhubungan dengan proses pengasuransian ulang kembali yang terhubung dari modul *underwriting*, modul klaim, dan modul *marketing*. Dalam reasuransi ini, terdapat ketentuan pembagian risiko atau *Spreading Of Risk (SOR)*. Dalam modul reasuransi, ditetapkan metode reasuransi sebagai berikut:

- a. Reasuransi Fakultatif, yang merupakan metode reasuransi ketika jumlah nominal atau ketentuan melebihi batas atau nasabah dan pihak reasuransi melakukan ketentuan diluar *treaty* dalam menentukan pembagiannya dan sifatnya *case by case* atau opsional.
- b. Reasuransi *Treaty*, yang merupakan metode reasuransi berdasarkan ketentuan-ketentuan yang telah disepakati antara pihak nasabah dengan

pihak reasuransi dan pihak perusahaan Askrindo dalam penentuan pembagian penanggungan tersebut pada perjanjian formal dan legal antara tertanggung ulang yang merupakan perusahaan reasuransi atau dengan tertanggung ulang lainnya.

Menurut ketentuan perusahaan Askrindo, terdapat beberapa jenis bisnis reasuransi yang antara lain:

1. Reasuransi *Outgoing*, merupakan bisnis reasuransi yang melibatkan perusahaan Askrindo sebagai pihak utama atau pemimpin dari asuransi tersebut
2. Reasuransi *Incoming*, merupakan bisnis reasuransi yang melibatkan perusahaan Askrindo sebagai pihak pengikut atau *following insurer*.

Berdasarkan kepala *software tester*, perusahaan Askrindo menentukan *treaty* sebagai berikut:

1. *Quota share*, merupakan *treaty* dimana perusahaan reasuransi yang telah menyepakati perjanjian tersebut wajib untuk memberikan atau menerima risiko sebagian dari perusahaan Askrindo yang telah ditetapkan. *Treaty* ini termasuk dalam bagian *proporsional*.
2. *Surplus*, merupakan *treaty* dimana perusahaan reasuransi yang telah menyepakati perjanjian tersebut wajib untuk memberikan jumlah risiko atau pertanggungan yang telah melebihi batas dari tanggungan perusahaan Askrindo. *Treaty* ini termasuk bagian *proporsional*.

3. *Facultative Obligatory*, merupakan *treaty* yang pembagian skema dari risiko tersebut seperti surplus tetapi terdapat beberapa ketentuan yang ditentukan dari pihak ketiga yang merupakan perusahaan reasuransi. *Treaty* ini termasuk dalam bagian *proporsional*.
4. *Excess of Loss*, merupakan *treaty* yang disepakati oleh perusahaan reasuransi dimana objek yang diasuransikan adalah hasil kerugian atau *losses* dari perusahaan Askrindo yang melebihi ketentuan dari perusahaan Askrindo, sehingga perusahaan reasuransi ikut menanggung sampai batas limit tertentu dari perusahaan reasuransi tersebut. *Treaty* ini termasuk dalam bagian *non proporsional*.
5. *Stoploss Cover*, merupakan *treaty* yang disepakati oleh perusahaan reasuransi dimana melindungi perusahaan Askrindo terhadap kerugian yang melebihi batas tertentu atau kerugian pada suatu *business* tertentu. *Treaty* ini termasuk dalam bagian *non proporsional*.
6. *Agregate Excess of Loss*, merupakan *treaty* yang disepakati oleh perusahaan reasuransi dimana mempunyai ketentuan yang sama seperti *treaty Stop Loss Cover* namun limit tersebut dinyatakan dalam suatu jumlah atau angka.

3.3.1.4 Modul Klaim

Modul klaim merupakan modul yang berfungsi untuk mengatur atau mengelola data terkait yang berhubungan klaim asuransi dari nasabah seperti

pengelolaan pemberian ganti rugi atas objek yang ditanggungkan oleh nasabah atau pihak ketiga. Pada modul ini dibutuhkan data yang telah ada dan diolah oleh modul marketing, *underwriting*, dan reasuransi. Proses utama pada modul klaim merupakan pencatatan atas pengajuan klaim, proses validasi data, survei terhadap pengajuan tersebut, sampai pengambilan keputusan untuk penyelesaian klaim atau *settlement*. Pengajuan klaim terhadap objek pertanggungan kepada perusahaan Askrindo dapat diajukan kepada semua kantor cabang.

Proses atau alur inti dari proses klaim dalam sebuah kantor cabang sebagai berikut:

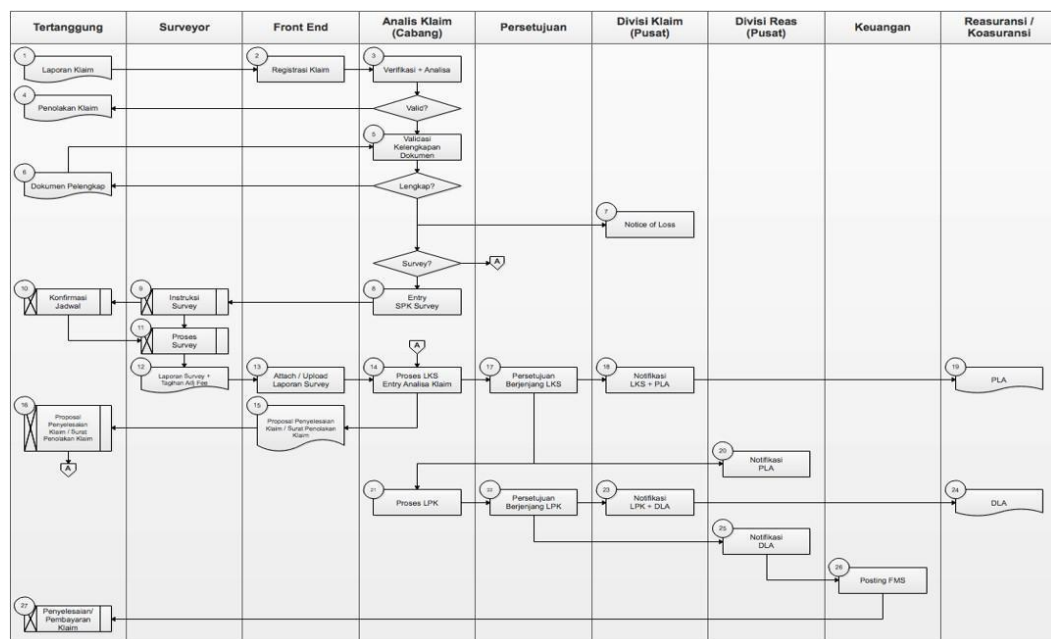
1. Nasabah melakukan pengajuan klaim dengan membuat laporan klaim terhadap salah satu kantor cabang perusahaan Askrindo.
2. Laporan tersebut akan diterima bagian depan atau *front end* untuk dikumpulkan dan langsung diteruskan kepada analis bagian klaim pada kantor cabang tersebut.
3. Pihak analis bagian klaim pada kantor cabang akan menganalisa laporan tersebut terkait dari sisi kesesuaian klaim, kelengkapan dokumen yang diperlukan untuk menganalisa klaim.
4. Apabila analis klaim menyatakan *valid* terhadap laporan tersebut, maka laporan NOL (*Notice Of Loss*) akan terbuat dan akan dikirimkan kepada divisi klaim kantor pusat. Apa bila ada koasuransi, maka dokumen NOL tersebut akan dikirimkan langsung kepada kantor pusat, dan notifikasi

dokumen tersebut kepada kantor cabang penerbit polis apabila penerbitan polis tersebut pada tempat yang berbeda.

5. Kemudian apabila diperlukan untuk meninjau tempat objek pertanggungan atau tempat lokasi kerugian tersebut sesuai dengan laporan klaim nasabah, maka dilakukan proses survei beserta pencatatan biaya dari survei tersebut, dan laporan dari hasil survei tersebut.
6. Kemudian akan terbuat dokumen LKS (Laporan Kerugian Sementara) yang berisi nilai kerugian awal yang diklaim oleh nasabah. Dokumen tersebut akan dibuat oleh sistem pada pengguna analis cabang klaim tersebut dan akan dilakukan persetujuan secara berjenjang hingga kepada divisi klaim pusat dengan ketentuan yang berlaku.
7. Setelah LKS disetujui, maka akan terbuat notifikasi dokumen LKS dan PLA (*Preliminary Loss Advice*) yang merupakan dokumen keterangan kerugian berdasarkan laporan klaim tersebut dan LKS yang dikirimkan kepada perusahaan reasuransi. Apabila ada perusahaan Koasuransi, maka PLA tersebut juga dikirimkan kepada perusahaan Koasuransi
8. Kemudian dokumen LPK (Laporan Penyelesaian Kerugian) terbuat berdasarkan LKS yang disetujui yang berisi nilai kerugian, biaya-biaya yang dibayarkan oleh pihak perusahaan Askrindo terhadap kerugian tersebut.

9. LPK akan dilakukan persetujuan berjenjang hingga divisi klaim pusat sesuai aturan yang berlaku

10. Setelah LPK disetujui, maka akan dilakukan pembuatan dokumen DLA (*Definite Loss Advice*) yang merupakan keterangan kerugian berdasarkan LPK yang dibuat dan ketentuan *treaty*, serta aturan berdasarkan pencatatan pada modul *business partner* dan modul reasuransi kepada koasuransi apabila ada.



Gambar 3.5. Diagram swimlane terkait workflow pengajuan klaim serta pihak yang terlibat

3.3.1.5 Test Scenario

Sebelum melakukan pengujian atau pengecekan dalam kegiatan kontrol kualitas sistem ACS, maka diperlukan untuk mempelajari skema pengujian dalam aplikasi tersebut. Skema pengujian terhadap sistem ACS, terdapat dalam sebuah

dokumen *test case* yang telah diberikan oleh kepala *software tester* untuk dipelajari.

	PRODUK	Priority	SKENARIO ID	DESKRIPSI	Effort Est. (minutes)	N	O	P	Q
1			TSCEN-UDWR-001	PROSES UNDERWRITING (Simple Approval - Level KaSie) - NORMAL CASE					
2	1	ASUM	TSCEN-UDWR-001#003	Akseptasi Sempel Polis	20				
4	1	ASUM	TSCEN-UDWR-001#004	Akseptasi Sempel Polis dengan Premi Full	20				
5	1	ASUM	TSCEN-UDWR-001#005	Akseptasi Sempel Polis dengan Premi Prorate	30				
6	1	ASUM	TSCEN-UDWR-001#006	Akseptasi Sempel Polis dengan Premi Scaling	30				
7	1	ASUM	TSCEN-UDWR-001#007	Akseptasi Koas In (Indirect) (Askrindo sebagai Member)	30				
8	1	ASUM	TSCEN-UDWR-001#008	Akseptasi Polis dengan KOAS (Askrindo sebagai Leader)	40				
9	1	ASUM	TSCEN-UDWR-001#009	Akseptasi Polis dengan Installment	40				
10	1	ASUM	TSCEN-UDWR-001#010	Akseptasi Polis dengan Perluasan	40				
11	1	ASUM	TSCEN-UDWR-001#011	Akseptasi Polis dengan REAS Facultative	40				
12	1	ASUM	TSCEN-UDWR-001#012	Akseptasi Polis dengan KOAS dan REAS Facultative	45				
13	2	ASUM	TSCEN-UDWR-001#013	Akseptasi Polis Single Objek dan Multiple Currency	30				
14	2	ASUM	TSCEN-UDWR-001#014	Akseptasi Polis dengan KOAS, REAS Facultative dan Multiple Objek	50				
15	2	ASUM	TSCEN-UDWR-001#015	KOAS, REAS Facultative, Multiple OBJEK dan Multiple Currency	50				
16	2	ASUM	TSCEN-UDWR-001#016	KOAS, REAS Facultative, Multiple OBJEK, Perluasan dan Multiple Currency	55				

Gambar 3.6. Cover dokumen *Test case* yang berisi *test scenario* untuk dipelajari

Dokumen *test case* tersebut terdiri dari kumpulan *test scenario* yang berisi langkah-langkah dalam pengujian sistem ACS tersebut dalam kondisi tertentu. *Test scenario* yang berbeda memiliki nomor unik untuk memudahkan ketika melakukan eksekusi langkah-langkah tersebut. *Test scenario* yang berada di dalam dokumen

test case ini, digunakan secara berulang dalam pengujian. Skenario dalam *test case* dikelompokkan sesuai dengan modul besar dari sistem ACS.

Scenario ID	Description	Precondition	Test Procedure / Step	Validation Element	Expected Result	Status	Severity	Defect ID	Remarks
TS-SCEN-UDWR-001	PROSES UNDERWRITING [Simple Approval - Level Kafe] - NORMAL CASE								
1	Login User Role - Front Desk		Buka menu Akseptasi untuk membuat Polis baru	User Interface	Sistem menampilkan form pembuatan polis baru	OK			
2			Input semua informasi yang diperlukan terutama yang bersifat mandatory pada tab General Info	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
3			Tambahkan Object pada tab Object Pertanggungan A. Input semua informasi object pertanggungan yang diperlukan, terutama yang bersifat mandatory B. Pada panel Risiko tekan tombol Hitung Premi untuk menghitung premi yang memiliki tipe Basic Cover (Mandatory) / tekan tombol " untuk menambah risiko yang memiliki tipe Perlakuan (Optional) C. Simpan informasi pada tab Obyek Pertanggungan	User Interface	- Hasil premi sesuai dengan metode perhitungan yang digunakan - Semua informasi yang terinput berhasil tersimpan sementara	OK			
4			Pada tab Klausula pilih Klausula dan Warranty	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
5			Jalankan fungsi hitung premi pada tab Summary	User Interface	Sistem akan melakukan perhitungan premi dan akan menampilkan ringkasan premi dan risikan nota premi	OK			
6			Pada panel Disposisi, input analisa atau disposisi terhadap polis tersebut	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK W/ NOTE			Cara pencarian dan cara penginputan Profil ID tertanggung, Cara pencarian dan penginputan Blok risiko masih belum sesuai

Gambar 3.7. Skenario langkah dalam melakukan *testing* kode

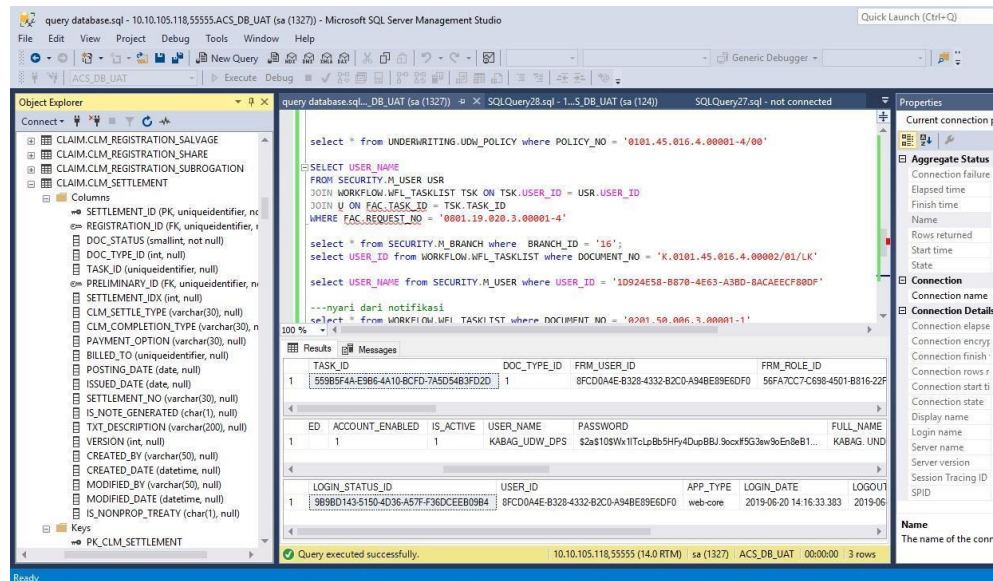
3.3.1.6 Database Server

Sebelum pengujian, diperlukan untuk mempelajari sistem basis data yang digunakan pada sistem ACS. Untuk menyimpan semua data yang telah dicatat kedalam sistem serta mengaksesnya kembali, sistem ACS menggunakan *Relational Database System* atau sistem basis data relasional (RDBMS) *Microsoft SQL Server*. Akses database dalam pengembangan dilakukan pada jaringan pribadi atau *private network* pada kantor PT Sigma Cipta Caraka. *Database* sistem ACS tersebut terdiri tabel-tabel yang dibuat berdasarkan diagram objek yang dirancang oleh *system analyst* dan kemudian diberikan kepada developer untuk mengerjakannya. Untuk melakukan pengetesan terhadap sistem ACS maka

diperlukan pemahaman atas struktur dan tabel *database* yang diperlukan dalam pengetesan, antara lain:

- a. INFORMATION_SCHEMA, berdasarkan dokumentasi *Microsoft SQL Server* merupakan *schema* yang menyimpan informasi terkait *database* lain yang ada di dalam satu server tersebut. INFORMATION_SCHEMA, menyimpan informasi terkait tabel, kolom, *constraint* atau aturan, relasi, serta tipe data dari setiap kolom yang berada pada tabel.
- b. WORKFLOW.WFL_TASKLIST, merupakan tabel yang berada pada *database* sistem ACS yang menyimpan informasi terkait alur dokumen yang mereferensi atau memiliki *foreign key* pada tabel *user*, serta tabel terkait jenis dokumen tersebut yang sedang aktif atau on *insert*.
- c. UNDERWRITING.UDW_POLICY, merupakan tabel yang berada pada sistem ACS yang menyimpan informasi terkait SPPA, dan polis ketika terbit.
- d. UNDERWRITING.UDW_POLICY_OBJECT, merupakan tabel sistem yang menampung informasi terkait objek yang ditanggungkan pada polis tersebut. Tabel ini mempunyai refrensi *key* atau *foreign key* pada tabel UNDERWRITING.UDW_POLICY, karena memiliki hubungan *one to many*.
- e. CLAIM.CLM_SETTLEMENT, merupakan tabel sistem yang menampung informasi terkait objek dokumen LPK.

- f. CLAIM.CLM_PRELIMINARY, merupakan tabel sistem yang menampung informasi terkait objek dokumen LKS



Gambar 3.8. Tampilan struktur tabel CLAIM.CLM_SETTLEMENT serta hasil dari *query* pencarian data pengguna berdasarkan tabel *login* dan tabel akun pengguna

3.3.2 Minggu Dua, Pembelajaran *Environment Deployment, Jenkins, Jira*, dan *deploy execute Query Database*

Kegiatan yang dilakukan pada minggu kedua adalah pengenalan sekaligus pembelajaran akan hal yang terkait dalam lingkungan pada pengembangan sistem, serta manajemen pengaturan tugas. Menurut kepala *software tester*, ada beberapa istilah yang harus dimengerti terlebih dahulu sebelum melakukan pengujian terhadap sistem ACS. Istilah-istilah tersebut merupakan:

- a. *Deployment*, yang merupakan sebuah aksi untuk melakukan *commit* dari pemutakhiran sistem berdasarkan *source code* yang diperbaharui, tampilan antar muka yang diperbaharui, *database* yang diperbaharui, serta hal lain yang terkait dalam sumber daya atau *resources* yang dibutuhkan oleh aplikasi tersebut kedalam lingkungan atau *environment* pengembangan tertentu.
- b. *Bug* atau *defect*, yang merupakan sebuah kesalahan baik proses, alur, atau tampilan dari sistem ACS yang tidak sesuai dengan ketentuan perusahaan Askrindo atau bisnis

Dalam pengembangan sistem ACS ada lima lingkungan pengembangan sistem untuk *deployment* agar mencegah adanya kesalahan atau *miss* dalam pemutakhiran sistem, diantaranya adalah:

1. Lokal

Environment lokal merupakan lingkungan dimana terdapat *commit* dilakukan oleh *developer* pada *server repository* atau arsip terkait source code, atau sumber daya yang dibutuhkan oleh sistem ACS.

2. *Staging*

Environment Staging merupakan lingkungan dimana *deployment* dilakukan ketika *developer* telah melakukan *deploy* pada server lokal dan telah diuji secara kasar dan sesuai dari *environment* lokal dan merupakan *environment software tester* melakukan pengujian terhadap fitur atau *bug* yang sudah diperbaiki.

3. *User Acceptance Test (UAT)*

Environment UAT merupakan lingkungan dimana *deployment* dilakukan ketika *software tester* telah menyetujui bahwa fitur atau *bug* tersebut telah diperbaiki secara benar pada *environment staging* dan *server* ini digunakan sebagai training dari karyawan perusahaan Askrindo yang menggunakan sistem ACS.

4. Migrasi

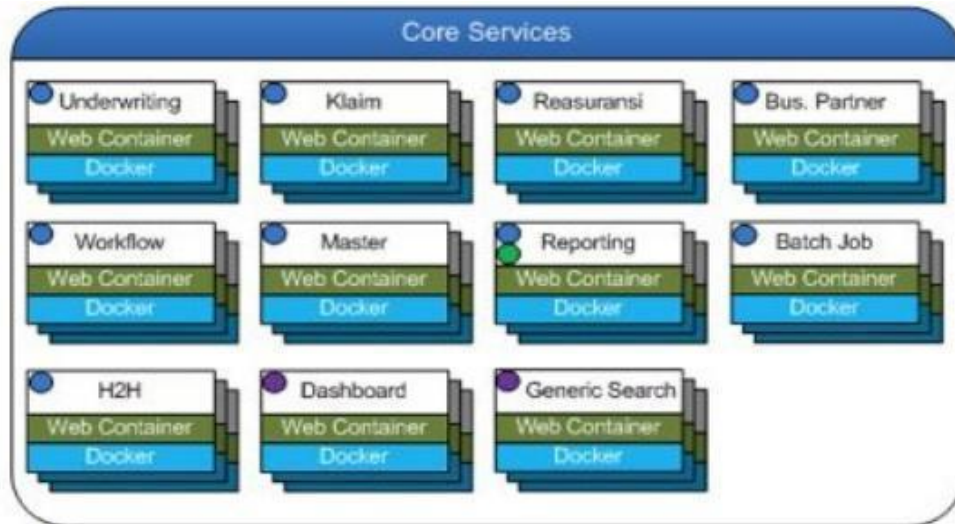
Environment migrasi merupakan lingkungan dimana *deployment* dilakukan apabila tester yang melakukan pelatihan kepada karyawan perusahaan Askrindo tidak menemui kendala dan tidak ada masukan dari pihak perusahaan Askrindo pada *environment* UAT. Kemudian, migrasi berfungsi sebagai tempat dimana tahap *software tester* melakukan pengujian terhadap sistem sesuai fitur atau bug yang telah diperbaharui dan dilakukannya *reproduce bug* apabila ada keluhan dari pihak perusahaan Askrindo pada *environment production*, dimana dalam melakukan pengujian menggunakan data transaksi asli yang sudah ada berdasarkan perusahaan Askrindo sebelumnya. *Deploy* pada migrasi dilakukan apabila ada *request* dari *system analyst* maupun *business analyst*.

5. *Production*

environment terakhir adalah *production*, dimana *deployment* dilakukan apabila fitur atau *bug* telah diperbaiki sebagaimana mestinya berdasarkan *environment* sebelumnya. Pada *environment* ini, tidak diperbolehkan untuk melakukan transaksi atau melakukan penambahan data, karena *environment* ini

merupakan sistem yang sedang berjalan dan dipakai dalam pekerjaan sehari-hari oleh perusahaan Askrindo

Setiap *environment* memiliki server secara terpisah baik *server* aplikasi, dan *server database*. Arsitektur sistem dibuat dan *di-deploy* ke dalam sebuah *environment* tertentu menganut konsep *microservices* dan menggunakan *framework Spring Boot* pada *back-end* dan menggunakan *framework Angular* dalam *front-end* dan setiap service yang dibuat, memiliki lingkungan sendiri (*sandbox*) yang diibaratkan sebagai sebuah *container* menggunakan *docker*. *Microservices* merupakan arsitektur dalam pengembangan sistem yang membagi sistem terdiri dari aplikasi-aplikasi kecil yang bekerja secara independen dan memiliki tanggung jawab secara spesifik, sehingga memudahkan dalam *deploy*, perluasan skala, serta pengujian (jurnal *microservice*). Sistem ACS memiliki modul utama yaitu *underwriting*, reasuransi, *business partner (marketing)* masing-masing dibagi menjadi sebuah *services* yang independen diikuti dengan modul yang lain dan saling mengirim data dalam konsep *microservice*.

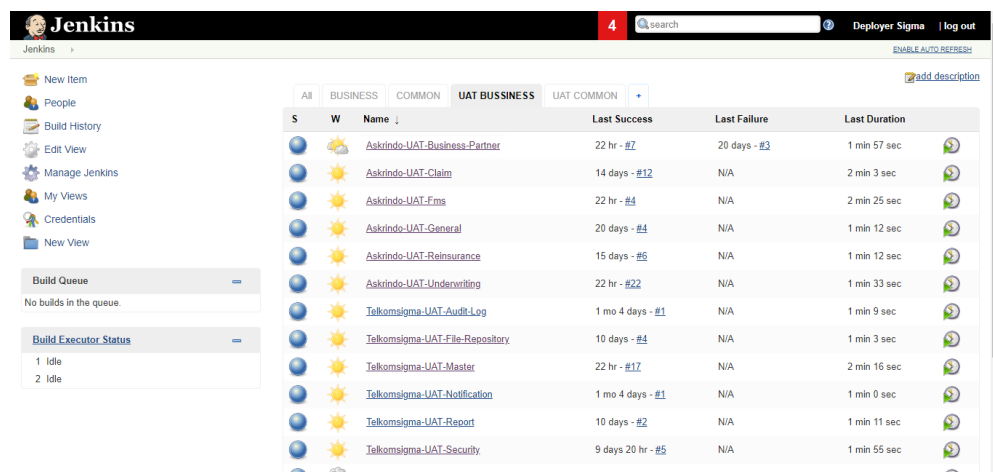


Gambar 3.9. Diagram *service* yang diperlukan pada sistem ACS yang menganut konsep *microservices*

Developer melakukan pembaharuan atau perubahan *service* setiap saat jika ada perubahan ketentuan atau *bug* dalam sistem ACS yang berkaitan dengan *service* tersebut. Setelah melakukan perubahan dalam *environment* lokal, maka diperlukan untuk dilakukan *deploy* kepada *environment staging* agar, *software tester* dapat melakukan pengujian terhadap bagian sistem atau fitur yang terhubung dalam *service* tersebut menggunakan *tools Jenkins*.

Berdasarkan dokumentasi resmi *tools* tersebut, *Jenkins* merupakan server yang dapat digunakan secara bebas (*open source*) yang berfungsi dalam mengatur pembuatan, pengujian, dan *deployment project*. Sesuai dengan konsep *microservices*, ketika *developer* melakukan pembaharuan sumber daya yang dibutuhkan aplikasi seperti *source code* yang telah dilakukan *commit* pada

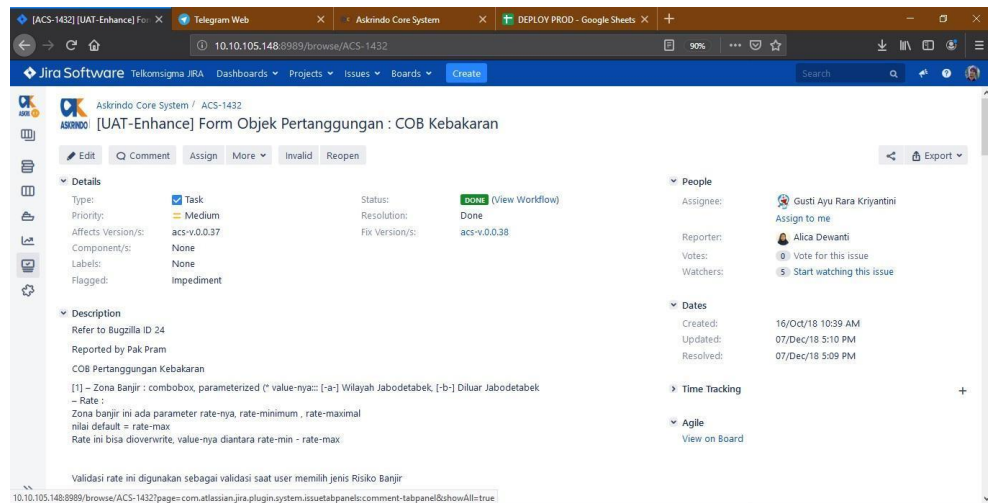
environment lokal, maka akan dilakukan *deploy* pada *server staging* dengan *tools Jenkins* khusus *server staging* terkait dengan pembaharuan *service* tertentu dilihat dengan catatan kode revisi dari developer. Begitu pula dengan *deploy* lingkungan berjenjang selanjutnya ke UAT, maka dilakukan pada *server* UAT dengan *service* terkait dengan kode revisi terkait.



S	W	Name	Last Success	Last Failure	Last Duration
		Askrindo-UAT-Business-Partner	22 hr - #7	20 days - #3	1 min 57 sec
		Askrindo-UAT-Claim	14 days - #12	N/A	2 min 3 sec
		Askrindo-UAT-Fms	22 hr - #4	N/A	2 min 25 sec
		Askrindo-UAT-General	20 days - #4	N/A	1 min 12 sec
		Askrindo-UAT-Reinsurance	15 days - #6	N/A	1 min 12 sec
		Askrindo-UAT-Underwriting	22 hr - #22	N/A	1 min 33 sec
		Talksigma-UAT-Audit-Log	1 mo 4 days - #1	N/A	1 min 9 sec
		Talksigma-UAT-File-Repository	10 days - #4	N/A	1 min 3 sec
		Talksigma-UAT-Master	22 hr - #17	N/A	2 min 16 sec
		Talksigma-UAT-Notification	1 mo 4 days - #1	N/A	1 min 0 sec
		Talksigma-UAT-Report	10 days - #2	N/A	1 min 11 sec
		Talksigma-UAT-Security	9 days 20 hr - #5	N/A	1 min 55 sec

Gambar 3.10. Tampilan *service* yang terdaftar dalam *server Jenkins* UAT yang dapat di-*deploy*

Selanjutnya, dilakukan pembelajaran terhadap *Jira software* yang ditugaskan oleh kepala *software tester*. Menurut kepala *software tester*, *Jira* merupakan *software* berbayar yang berguna dalam manajemen proyek dalam hal pemberian tugas, pelaporan *bug*, serta analisa waktu yang diperlukan dalam mengerjakan suatu pekerjaan atau modul hingga sistem secara keseluruhan. Pada pembelajaran ini dan untuk penggunaan dalam pengujian, digunakan akun kepala *software tester* dikarenakan lisensi yang terbatas untuk tiap pengguna *Jira*.



Gambar 3.11. Tampilan sebuah *task* yang telah dibuat berdasarkan laporan dari *business analyst* pada *Jira*

Dalam proyek perusahaan Askrindo, *developer* dan *software tester* menerima tugas atau *issue* dari *business analyst*, *system analyst*, dan kepala *software tester*. *Issue* tersebut memiliki tipe yang berbeda, diantaranya yang dipelajari adalah:

1. Task

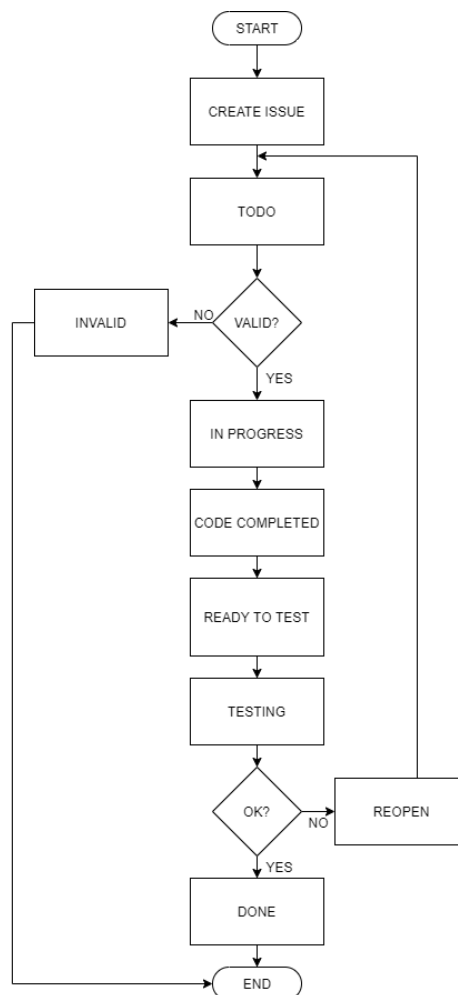
Merupakan pekerjaan yang diberikan oleh pembuat *issue Jira*. Setiap *task* yang terbuat bisa saja memiliki relasi dengan *task* lain. *Task* memiliki status *todo*, *inprogress*, *code completed*, dan *done*.

2. Bug

Merupakan temuan malfungsi atau *defect* dalam sistem ACS yang dibuat selain *developer*. Setiap *bug* yang terbuat, bisa saja memiliki relasi dengan *issue*

bug yang lain. *Bug* memiliki status *valid*, *invalid*, *todo*, *inprogress* *code completed*, *ready to test*, *testing*, *reopen*, *done*.

Setelah mempelajari *tools Jira* dalam membuat *issue*, dan menganalisa sebuah *issue*, maka dilakukan pembelajaran mengenai siklus *bug* atau *bug lifecycle*. *Bug lifecycle* merupakan sebuah alur atau tahap-tahap yang digambarkan secara berulang. Berikut terdapat gambar diagram yang menampilkan alur *bug lifecycle*.



Gambar 3.12. Diagram alir yang menggambarkan *bug lifecycle*

Berdasarkan gambar 3.12, terdapat tujuh status sekaligus proses yang dijadikan status dari *Jira* sebagai dasar dalam *bug lifecycle*. Berikut penjelasan dari status proses tersebut:

1. *Todo*

Todo merupakan status yang menunjukkan *issue bug* yang telah terbuat sedang menjalani proses analisa atau persiapan.

2. *Invalid*

Invalid merupakan status yang menunjukkan *bug* yang telah dibuang *issue*, sedang menjalani proses penutupan karena tidak sesuai atau tidak benar

3. *In Progress*

Merupakan status yang menunjukkan *bug* tersebut sedang menjalani proses perbaikan, penambahan dalam sumber daya sistem ACS atau proses yang terkait dalam memenuhi *issue* tersebut.

4. *Code Completed*

Merupakan status yang menunjukkan *bug* tersebut telah diselesaikan oleh *developer* terkait dengan pembaharuan versi berdasarkan masalah pada *issue* tersebut.

5. *Ready To Test*

Merupakan status yang menunjukkan *bug* tersebut sudah dipersiapkan agar *software tester* melakukan *testing*.

6. *Testing*

Merupakan status yang menunjukkan *bug* tersebut sedang dilakukan pengujian.

7. *Reopen*

Status yang menunjukkan bahwa *bug* tersebut tidak terselesaikan dan masih mengalami malfungsi terhadap sistem, sehingga perlu dilakukan perbaikan ulang kepada *developer*.

8. *Done*

Status yang menunjukkan bahwa *bug* tersebut telah diperbaiki dan telah melewati pengujian oleh *software tester*.

Setelah mempelajari *bug lifecycle*, kepala *software tester* menugaskan untuk melanjutkan pekerjaan dengan tim pada proyek perusahaan Askrindo tersebut dengan melakukan pembuatan *query* dan melakukan eksekusi *query* berdasarkan diskusi dengan *developer* terkait dengan *issue Jira* yang diminta oleh *system analyst* pada *environment migrasi*. *Deployment* pada migrasi dilakukan untuk dipersiapkan agar dapat dilakukan pengujian dan apabila berhasil dilakukan *deploy* pada *environment production* ketika semua *issue* yang diuji sudah baik atau lolos

oleh *software tester*. Untuk *deploy migrasi* ditentukan pak Krishna Putra selaku *system analyst*. Kode *issue Jira* tersebut adalah ACS-3937 dan ACS-4100.

```
1 ALTER TABLE UNDERWRITING.UDW_POLICY ADD IS_VAT_COLLECTED CHAR(1);
2 ALTER TABLE UNDERWRITING.UDW_SUMMARY_DIRECT ADD AMT_DPP DECIMAL(18,2);
3 ALTER TABLE UNDERWRITING.UDW_SUMMARY_DIRECT ADD AMT_DPP_DELTA DECIMAL(18,2);
4 ALTER TABLE UNDERWRITING.UDW_SUMMARY_DIRECT ADD AMT_DPP_RATE DECIMAL(18,2);
5 ALTER TABLE UNDERWRITING.UDW_SUMMARY_DIRECT ADD AMT_DPP_IDR DECIMAL(18,2);
6
7 ALTER TABLE UNDERWRITING.UDW_SUMMARY_DIRECT_DT ADD AMT_DPP DECIMAL(18,2);
```

Gambar 3.13. Script SQL untuk deploy ACS-3937

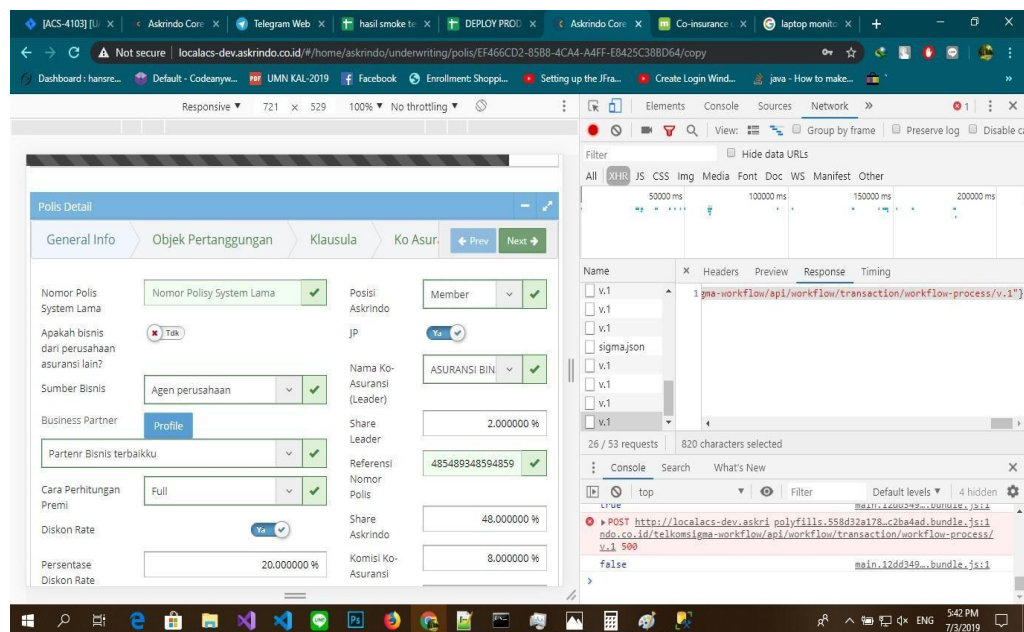
Pada melakukan *deploy environment* migrasi dengan kode *Jira* ACS-3937, dilakukan *deploy* pembaharuan pada struktur skema tabel *UNDERWRITING* dengan menambahkan kolom *flag* atau tanda PPN telah terkumpul, serta kolom terkait dengan nominal dasar pengenaan pajak. Kemudian untuk kode ACS-4100 dilakukan *deploy* penambahan data pada tabel *general ledger* (GL) terkait id GL per produk, dan keterangan sesuai *system analyst* menggunakan *statement INSERT*.

```
Run Terminal Help ACS-4100.sql - Visual Studio Code [Unsupported]
ACS-4100.sql X
ents > magang > pak Zein > ACS SCRIPT > ACS-4100.sql
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0101','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0102','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0103','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0104','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0105','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0106','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0201','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0202','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0203','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0204','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0205','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0301','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0302','direct_commission_vat_collected','2.02.03.03.001.001'
MODIFIED_DATE,ERP_JOURNAL_LINES_TRANS_SOURCE_ID) values (NEWID(),'0303','direct_commission_vat_collected','2.02.03.03.001.001'
```

Gambar 3.14. Script SQL untuk deploy ACS-4100

3.3.3 Minggu Tiga, *Testing* Pada Modul *Underwriting* Reasuransi, dan *Deploy* *Execute Query Database*

Pada minggu ketiga, dilakukan pengujian lanjutan pada sistem ACS pada bagian modul *Underwriting*. Pengujian ini dilakukan dengan tujuan untuk memastikan bahwa proses *copy polis* telah berjalan dengan lancar setelah *deploy* terakhir. Namun ketika dilakukan pengujian *copy polis* pada salah satu *user data entry* kantor cabang, polis tersebut tidak dapat disimpan dan tidak terdapat *feedback* apapun.



Gambar 3.15. Terdapat *Error 500* ketika melakukan *save* polis pada *environment* migrasi

1 Back To Cover										Askrindo Core System - Test Scenario																																																											
4 Scenario ID		TSCEN-UDWR-001										Scenario Last Modified Date:																																																									
5 Description		PROSES UNDERWRITING (Simple Approval - Level KaSie) - NORMAL CASE																																																																			
6 N		Precondition		Test Procedure / Step				Validation Element		Expected Result				Status		Severity		Defect ID		Remarks																																																	
380		#022 Akseptasi Simpel Polis Copy All																																																																			
381		1 - Login User Role : Front Desk - Status Polis = Terbit		Buka menu Akseptasi untuk memilih polis berstatus terbit yang akan di Copy				User Interface		- Sistem menampilkan form untuk pembuatan polis dari hasil copy polis yang ada - Semua data dan informasi dari polis awal ter-copy ke polis yang				OK																																																							
382		2		Lakukan perubahan yang diinginkan pada tab General Info dan pastikan semua informasi yang bersifat mandatory sudah terisi				User Interface		Semua informasi yang terinput berhasil tersimpan sementara				OK																																																							
383		3		Lakukan perubahan yang diinginkan pada tab Object Pertanggungan atau tambahkan Object pada tab Object Pertanggungan A. Input semua informasi object pertanggungan yang diperlukan terutama yang bersifat mandatory B. Pada panel Risiko tekan tombol Hitung Premi untuk menghitung premi yang memiliki tipe Basic Cover (Mandatory) / tekan tombol "+" untuk menambah risiko yang memiliki tipe Perluasan (Optional). C. Simpan informasi pada tab Obyek Pertanggungan				User Interface		- Nilai Premi sesuai dengan metode perhitungan yang digunakan - Semua informasi yang terinput berhasil tersimpan sementara				OK																																																							
384		4		Pada tab Klausula pilih Klausula dan Warranty atau lakukan perubahan yang diinginkan				User Interface		Semua informasi yang terinput berhasil tersimpan sementara				OK																																																							
385		5		Jalankan fungsi hitung premi pada tab Summay bila ada perubahan pada nilai premi di object pertanggungan				User Interface		Sistem akan melakukan perhitungan premi dan akan menampilkan ringkasan premi dan rincian nota				OK																																																							
386		6		Pada panel Disposisi, input kembali analisa atau disposisi terhadap polis tersebut				User Interface		Semua informasi yang terinput berhasil tersimpan sementara				OK																																																							
Master-Klaim										Report_2018_10_29										User Approval										TSCEN-UDWR-001										TSCEN-UDWR-002										TS ... + : <																			

Gambar 3.16. Dokumen Test case proses copy polis yang dijalankan

Kemudian dilakukan analisa terkait dengan respon dari *code* aplikasi yang telah disimpan dalam *cache* komputer dengan membuka *console developer tools* yang berada dalam *browser Google Chrome*. Dalam bagian *console*, terdapat *error* yang dikirimkan dengan metode POST dan menampilkan kode 500. Berdasarkan dari hasil tersebut, dapat diartikan bahwa *error* ini diakibatkan pada bagian *back-end*. Kemudian dilakukan ekstrak data respon balik dari *back-end* tersebut yang berupa JSON (*JavaScript Object Notation*) untuk dianalisa.

```

{
  "timestamp": "03/07/2019",
  "status": 500,
  "error": "Internal Server Error",
  "exception": "Id.co.telkomsigma.workflow.exceptions.WorkflowException",
  "message": "status 500 reading Policyfeign#savePolis(String,String): content:\n{\"timestamp\":\"03/07/2019\",\"status\":500,\"error\":\"Inter",
  "path": "/telkomsigma-workflow/api/workflow/transaction/workflow-process/v.1"
}

```

Gambar 3.17. JSON yang merupakan respon dari back-end

Setelah menganalisa JSON yang telah diekstrak dengan melihat pesan pada indeks *array* JSON yang bernama *message*, terdapat masalah objek yang habis dari

masa sesi tersebut tersebut ketika menjalankan fungsi `savePolis()` pada *service* modul *underwriting* karena ada indikasi pesan *error* “*detached entity passed to persist*”. Kemudian dilakukan pemberitahuan kepada kepala *software tester* lain terkait hal tersebut, dan kepala *software tester* untuk mengkomunikasikan hal tersebut dengan *developer* yang terkait, serta menugaskan *software tester* lain untuk melakukan *reproduce* dari proses yang telah dilakukan. Hasil dari *reproduce* *software tester* lain memiliki *error* yang sama sehingga dapat disimpulkan proses copy polis memang tidak bisa dilakukan, sehingga *developer* akan melakukan fix terhadap proses tersebut.

Kemudian dilakukan eksekusi *query* berdasarkan diskusi dengan *developer* terkait dengan *issue Jira* yang pada *environment migrasi* berdasarkan permintaan *request* untuk *deploy* pada *environment production* pada minggu ini sesuai permintaan *system analyst*. Kode *Jira* yang dieksekusi tersebut adalah:

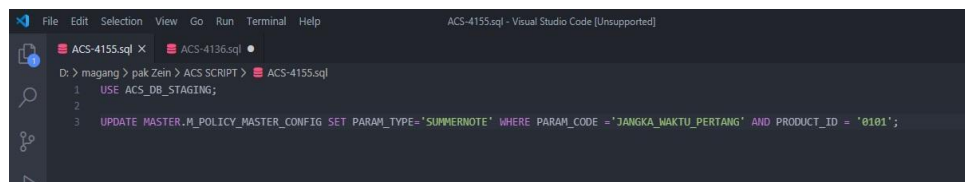
1. ACS-4136

```
1  USE ACS_DB_STAGING;
2
3  /* 2104 - Stop Usaha - gempa tsunami */
4  UPDATE [MASTER].[M_BUSINESS_PARAMETER]
5  SET [PARAMETER_VALUE] = '[{"ordering":1,"benefit":"Santunan untuk memulai usaha","hp":2500000,"currency":"IDR"}]'
6  WHERE [BUSINESS_PARAMETER_ID] = '75280103-3A53-4BEE-8D83-9D91E905C1B7';
7
8  /* 2103 - Stop Usaha Erupsi */
9  UPDATE [MASTER].[M_BUSINESS_PARAMETER] SET
10 [PARAMETER_VALUE] = '[{"ordering":1,"benefit":"Santunan untuk memulai usaha","hp":2500000,"currency":"IDR"}]'
11 WHERE [BUSINESS_PARAMETER_ID] = '4869BCF8-F8C1-4577-AACC-CD7A55395868';
12
13 /* 2105 - Asuransiku */
14 UPDATE [MASTER].[M_POLICY_WARRANTY_TYPE]
15 SET [DEFAULT_WARRANTY_TYPE] = 'Meninggal Dunia Akibat Kecelakaan dan atau Cacat Tetap Keseluruhan'
16 WHERE [POLICY_CONDITION_ID] = '6A29B27E-EC19-45AC-9655-E206E4FE6297';
```

Gambar 3.18. Script SQL untuk *deploy* ACS-4136

Pada gambar 3.17, merupakan *script* yang telah dibuat berdasarkan hasil diskusi dengan *developer* terkait ACS-4136 yang memperbaharui terkait dari parameter ketentuan dalam penambahan partner bisnis baru untuk modul *marketing*. Kemudian dilakukan perubahan deskripsi tipe waranti pada tabel master data kebijakan waranti.

2. ACS-4155



Gambar 3.19. Script Query ACS-4155

Pada gambar 3.19, merupakan *script* yang telah dibuat berdasarkan hasil diskusi dengan *developer* terkait ACS-4155 yang merupakan *bug* terhadap *field* jangka waktu pertanggungan dalam pengaturan *administrator* yang merupakan *field form* biasa yang perlu dirubah menjadi *summernote*. Maka dilakukan pembuatan *query* update pada tabel konfigurasi master regulasi dengan mengubah data parameter khusus TOC Harga benda (kode *product_id* = 0101).



Gambar 3.20. Field yang telah diubah menjadi summernote

313	#046	Create System Parameter				
314	1	Buka menu Master, lalu pilih sub menu System Parameter	User Interface	Sistem akan menampilkan halaman System Parameter	OK	
315	2	Klik tombol + untuk membuat data System Parameter baru	User Interface	Sistem akan menampilkan form System Parameter	OK	
316	3a	Input semua informasi yang diperlukan terutama yang bersifat mandatory pada form System Parameter	Database	Sistem menampilkan halaman konfirmasi proses penambahan data (YES/ NO)	OK	
317	3a1	Lalu lakukan proses Simpan data	Database	Sistem berhasil melakukan proses tambah data System Parameter ke Database	OK	
318	3a2	Konfirmasi proses tambah data (NO)	Database	Sistem batal melakukan proses tambah data System Parameter ke Database	OK	
319	3b	Input semua informasi yang diperlukan terutama yang bersifat mandatory pada form System Parameter	Database	Data System Parameter baru tidak berhasil ditambahkan ke Database	OK	
320	3b1	Lalu lakukan proses Pembatalan Simpan data	Database			
321	#047	Update System Parameter				
322	1	Buka menu Master, lalu pilih sub menu System Parameter	User Interface	Sistem akan menampilkan halaman System Parameter	OK	
323	2	Pilih data System Parameter yang akan di update pada halaman inquiry dengan	User Interface	Sistem berhasil menampilkan satu atau beberapa	OK	

Gambar 3.21. Dokumen *Test case* yang digunakan untuk melihat summer note pada konfigurasi master

3. ACS-4153

```

File Edit Selection View Go Run Terminal Help • ACS-4153.sql - Visual Studio Code (Unsupported)
D:\> msagang > pak Zein > ACS SCRIPT > ACS-4153.sql
1 UPDATE MASTER.M_POLICY_MASTER_CONFIG SET ROW_POS=ROW_POS+2 WHERE PARAM_CODE IN (
2 'RESIKO SENDIRI'
3 'KLAUSULA WARRANTIES',
4 'SUKU PREMI',
5 'PERHITUNGAN PREMI',
6 'CLAUSES',
7 'WARRANTIES',
8 'SUBJECTIVITY',
9 'ADDITIONAL_EXC'
10 ) AND PRODUCT_ID = '0101';
11
12 INSERT INTO MASTER.M_POLICY_MASTER_CONFIG
13 (CONFIG_ID, PRODUCT_ID, PARAM_CODE, PARAM_LABEL,
14 PARAM_LABEL_ENG, PARAM_TYPE, DATA_FORMAT, DEFAULT_VALUE,
15 REQUIRED, MIN_LENGTH, MAX_LENGTH, INVALID_VALUE_MSG, REQUIRED_VALUE_MSG,
16 MIN_VALUE, MAX_VALUE, FIELD_INDEX, ROW_POS, COLUMN_POS, DEPENDENCY_INPUT, DEPENDENCY_FIELD,
17 OPTION_VALUES, STATUS, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE)
18 VALUES(NEWID(), '0101', 'JAMINAN POKOK', 'Jaminan Pokok', 'TEXTFIELD', NULL, NULL, 1, NULL, NULL,
19 NULL, NULL, NULL, NULL, (SELECT ROW_POS+1 FROM MASTER.M_POLICY_MASTER_CONFIG WHERE PARAM_CODE='TOTAL_HARGA PERTANGG'
20 AND PRODUCT_ID='0101'), 1, NULL, NULL, 1, NULL, NULL, NULL, NULL);
21
22 INSERT INTO MASTER.M_POLICY_MASTER_CONFIG
23 (CONFIG_ID, PRODUCT_ID, PARAM_CODE, PARAM_LABEL, PARAM_LABEL_ENG, PARAM_TYPE, DATA_FORMAT, DEFAULT_VALUE, REQUIRED, MIN_LENGTH, MAX_LENGTH,
24 INVALID_VALUE_MSG, REQUIRED_VALUE_MSG, MIN_VALUE, MAX_VALUE, FIELD_INDEX, ROW_POS, COLUMN_POS, DEPENDENCY_INPUT, DEPENDENCY_FIELD,
25 OPTION_VALUES, STATUS, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE)
26 VALUES(NEWID(), '0101', 'P_JAMINAN PERLUASAN', 'Jaminan Perluasan', 'Jaminan Perluasan', 'TEXTFIELD', NULL, NULL, 1, NULL, NULL, NULL,
27 NULL, NULL, NULL, NULL, (SELECT ROW_POS+2 FROM MASTER.M_POLICY_MASTER_CONFIG WHERE PARAM_CODE='TOTAL_HARGA PERTANGG' AND PRODUCT_ID='0101'),
28 1, NULL, NULL, NULL, 1, NULL, NULL, NULL, NULL);

```

Gambar 3.22. Script SQL untuk *deploy* ACS-4153

Pada gambar 3.22, merupakan *script* yang telah dibuat berdasarkan hasil diskusi dengan *developer* terkait ACS-4153 yang merupakan *issue* untuk menambah sebuah ketentuan baru dalam konfigurasi aplikasi *administrator* atau

admin pada sistem ACS yang berupa field. Dilakukan *query insert* pada tabel konfigurasi master regulasi.

4. ACS-4158

```
D:\> magang > pak Zein > ACS SCRIPT > ACS-4158.sql
1  DELETE FROM MASTER.M_POLICY_MASTER_CONFIG WHERE PRODUCT_ID = '0101';
2
3  INSERT INTO MASTER.M_POLICY_MASTER_CONFIG
4  (CONFIG_ID, PRODUCT_ID, PARAM_CODE, PARAM_LABEL, PARAM_LABEL_ENG, PARAM_TYPE, DATA_FORMAT, DEFAULT_VALUE, REQUIRED, MIN_LENGTH, MAX_LENGTH, INVA
5  VALUES('25DF98CC-C0B5-4B76-9E90-C160263983AC', '0101', 'JENIS_JAMINAN', 'Jenis Jaminan', 'Type Of Coverage', 'SUMMERNOTE', NULL, NULL, 1, NULL, '
6  INSERT INTO MASTER.M_POLICY_MASTER_CONFIG
7  (CONFIG_ID, PRODUCT_ID, PARAM_CODE, PARAM_LABEL, PARAM_LABEL_ENG, PARAM_TYPE, DATA_FORMAT, DEFAULT_VALUE, REQUIRED, MIN_LENGTH, MAX_LENGTH, INVA
8  VALUES('DF804435-AC89-4028-98AA-A39C8480E125', '0101', 'JAMINAN_PERLUASAN', 'Jaminan Perluasan', 'Extention Coverage', 'SUMMERNOTE', NULL, NULL,
9  INSERT INTO MASTER.M_POLICY_MASTER_CONFIG
10 (CONFIG_ID, PRODUCT_ID, PARAM_CODE, PARAM_LABEL, PARAM_LABEL_ENG, PARAM_TYPE, DATA_FORMAT, DEFAULT_VALUE, REQUIRED, MIN_LENGTH, MAX_LENGTH, INVA
11 VALUES('C7D438B2-5586-4F14-857E-7B21AFD43535', '0101', 'NAMA_TERTANGGUNG', 'Nama Tertanggung', 'The Insured', 'SUMMERNOTE', NULL, NULL, 1, NULL,
12 INSERT INTO MASTER.M_POLICY_MASTER_CONFIG
13 (CONFIG_ID, PRODUCT_ID, PARAM_CODE, PARAM_LABEL, PARAM_LABEL_ENG, PARAM_TYPE, DATA_FORMAT, DEFAULT_VALUE, REQUIRED, MIN_LENGTH, MAX_LENGTH, INVA
14 VALUES('B7ED8E45-D060-4427-90C5-FA3D16E23E92', '0101', 'ALAMAT_TERTANGGUNG', 'Alamat Tertanggung', 'Insured Address', 'SUMMERNOTE', NULL, NULL,
15 INSERT INTO MASTER.M_POLICY_MASTER_CONFIG
16 (CONFIG_ID, PRODUCT_ID, PARAM_CODE, PARAM_LABEL, PARAM_LABEL_ENG, PARAM_TYPE, DATA_FORMAT, DEFAULT_VALUE, REQUIRED, MIN_LENGTH, MAX_LENGTH, INVA
17 VALUES('2CF51375-6C99-48F7-84D4-A86817BC8A52', '0101', 'JANGKA_WAKTU_PERTANG', 'Jangka Waktu Pertanggungan', 'Period Of Insurance', 'SUMMERNOTE'
```

Gambar 3.23. Script SQL untuk *deploy* ACS-4158

Pada gambar 3.23, merupakan *script* yang telah dibuat berdasarkan hasil diskusi dengan *developer* terkait. ACS-4158 yang merupakan *issue* untuk melakukan update master konfigurasi *administrator*, maka dilakukan reset data karena telah terjadi ada data yang *corrupt*, sehingga ACS-4153 dilakukan *query* kembali bersama, dengan ACS 4158 terkait dengan reset data master konfigurasi *admin*.

5. ACS-4186

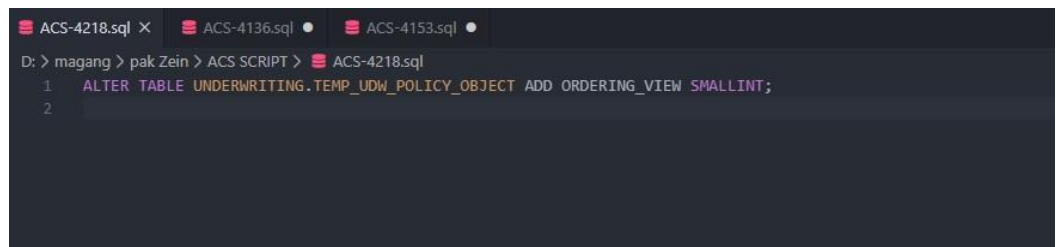
```
File Edit Selection View Go Run Terminal Help
ACS-4155.sql ACS-4153.sql ACS-4158.sql ACS-4186.sql X ACS-4136.sql
D:\> magang > pak Zein > ACS SCRIPT > ACS-4186.sql
1  ALTER TABLE REINSURANCE.REI_FAC_OUT_PANEL ALTER COLUMN TXT_SUBJECTIVITY varchar(5000);
```

Gambar 3.24. Script SQL untuk *deploy* ACS-4186

Pada gambar 3.24, merupakan *script* yang telah dibuat berdasarkan hasil diskusi dengan *developer* terkait. ACS-4186 yang merupakan *issue* untuk

melakukan perubahan *field* pada pengaturan klausula reasuransi pada modul Reasuransi. *Field* tersebut digunakan untuk menuliskan ketentuan klausula dalam polis yang terkena reasuransi fakultatif, namun terhitung *bug*, karena *field* tersebut terlalu sedikit yang sebelumnya hanya dapat menampung 1000 karakter, sehingga dilakukan update aturan dari kolom subjektivitas.

6. ACS-4218



Gambar 3.25. Script SQL untuk *deploy* ACS-4218

Pada gambar 3.25, merupakan *script* yang telah dibuat berdasarkan hasil diskusi dengan *developer* terkait. ACS-4218 yang merupakan *issue* untuk melakukan perubahan agar setiap polis yang memiliki objek pertanggung di-*entry* dengan cara *upload* dokumen yang berformat *excel*, sehingga mempunyai urutan objek yang berurutan dan dinamis. Oleh karena itu dilakukan *execute query script* pada tabel objek pertanggung polis sementara yang agar menambah kolom *ordering* sebagai *flag* atau penanda urutan.

Lalu, dilakukan pengujian terhadap *field Subjectivity* pada bagian modul reasuransi ketika polis baru telah disetujui, sebelum diterbitkan pada modul *underwriting* sesuai dengan kode *Jira* ACS-4186. *Field* tersebut digunakan untuk

menuliskan ketentuan klausula yang nantinya akan tertampil dalam *facultative reinsurance slip* sebagai dokumen yang akan dikirimkan pada perusahaan reasuransi, bagi polis yang terkena reasuransi fakultatif. Test ini dilakukan setelah sebelumnya telah dilakukan *deploy* pada *environment* migrasi baik dari aplikasi, maupun *database*.

The screenshot displays a web application interface for editing a reinsurance policy. The browser address bar shows the URL: `localhost:dev-askrindo.co.id/#/home/askrindo/reinsurance-editor/`. The interface includes several input fields and a table:

- Deductible:** A text area containing "Burglary & Theft : 5 % of claim minimum IDR 10000000 for".
- NET RATE:** A dropdown menu with "Tak" selected.
- Subjectivity:** A text area containing "pada saat input data facultative Testing field subjectivity pada saat input data facultative Testing field subjectivity pada saat input data facultative Testing field subjectivity".
- Catatan:** A text area containing "Catatan".
- Installment:** A section with a "Generate Installment" button and a table below it.

No.	Installment Amou	Due Date	Grace Period / PWI	Extended	No Nota

Gambar 3.26. Field polis yang sudah disetujui dalam akun pengguna divisi bagian reasuransi

Scenario ID	TSCEN-UDWR-001				Scenario Last Modified Date:			
Description	PROSES UNDERWRITING (Simple Approval - Level Kasie) - NORMAL CASE							
N	Precondition	Test Procedure / Step	Validation Element	Expected Result	Status	Severity	Defect ID	Remarks
180	1011 Akseptasi Polis dengan REAS Facultative							
	1 Login User Role : Front Desk	Buka menu Akseptasi untuk membuat Polis baru	User Interface	Sistem menampilkan form pembuatan polis baru	OK			
181	2	Input semua informasi yang diperlukan terutama yang bersifat mandatory pada tab General Info	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
182	3	Tambahkan Object pada tab Object Pertanggungan A. Input semua informasi object pertanggungan yang diperlukan, terutama yang bersifat mandatory B. Pada panel Risiko tekan tombol Hitung Premi untuk menghitung premi yang memiliki tipe Basic Cover (Mandatory) / tekan tombol "+" untuk menambah risiko yang memiliki tipe Perluasan (Optional). C. Simpan informasi pada tab Obyek Pertanggungan	User Interface	- Hasil Premi sesuai dengan metode perhitungan yang digunakan - Semua informasi yang terinput berhasil tersimpan sementara	OK			
183	4	Pada tab Klausula pilih Klausula dan Warranty	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
184	5	Jalankan fungsi hitung premi pada tab Summary	User Interface	Sistem akan melakukan perhitungan premi dan akan menampilkan ringkasan premi dan rincian nota	OK W/ NOTE			data lama yg di delete masih terhitung di tab reas facultative dan summary
185	6	Pada panel Disposisi, input analisa atau disposisi terhadap polis tersebut	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
186	7	Kirimkan polis ke Analis Underwriting untuk menjalankan workflow persetujuan polis dan polis juga dikirimkan ke divisi ReAsuransi	Database	- Semua input yang dilakukan dari step no 1 s/d 7 berhasil tersimpan di database - No Req SPPA terbentuk sesuai dengan format: (KodeCORB)(KodeCabang)(NoUrut)	OK			
◀ ... Master-Klaim Report_2018_10_29 User Approval TSCEN-UDWR-001 TSCEN-UDWR-002 Ts ... ➡								

Gambar 3.27. Dokumen Testcase yang digunakan untuk melihat field Subjectivity

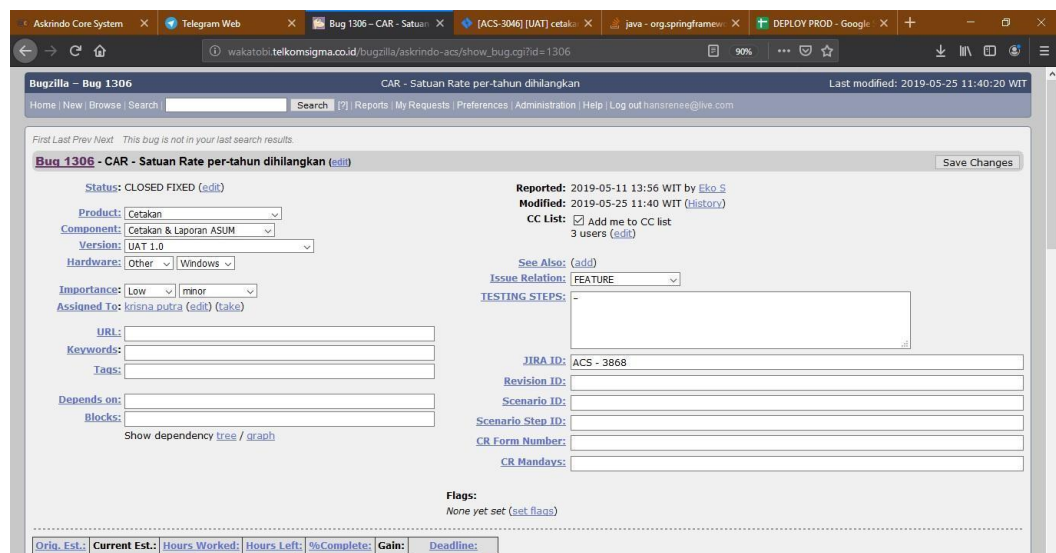
3.3.4 Minggu Empat, Pembelajaran *Bugzilla*, Testing Modul Underwriting

Upload dokumen, dan Deploy Execute Query

Pada minggu keempat, dilakukan pembelajaran terhadap *tools* manajemen *bug* yang bernama *bugzilla*. Menurut kepala *software tester*, *bugzilla* merupakan *tools* manajemen *bug* untuk menghubungkan pengguna sistem dari pihak perusahaan Askrimdo dengan tim pengembang sistem ACS pada PT Sigma Cipta Caraka. *Tools ini* berbeda dengan *Jira* dimana, *Jira* merupakan manajemen proyek secara keseluruhan serta hanya dikhususkan untuk tim pengembang sistem ACS pada PT Sigma Cipta Caraka. Dilakukan pembelajaran ini terhadap *bugzilla* terkait status dari *bug* tersebut yang diantaranya menurut kepala *software tester* adalah:

1. NEW, merupakan status *bug* telah dibuat secara baru.
2. ASSIGNED, merupakan status *bug* yang menunjukkan ada yang sedang melakukan analisa.

3. *REOPENED*, merupakan status *bug* yang telah diuji dan dilakukan perbaikan, namun hasil pengujian tersebut, *bug* masih ada.
4. *RESOLVED*, merupakan status *bug* yang telah diuji oleh pihak tim pengembang proyek namun tidak terjadi.

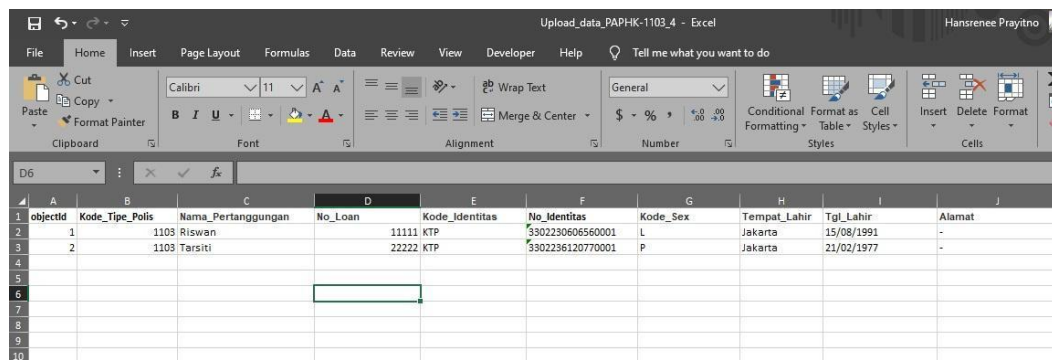


Gambar 3.28. Tampilan *bug* 1306 yang sudah dilakukan pembetulan dan telah sesuai berdasarkan pihak Perusahaan Askrindo

Kemudian dilakukan pengujian lanjutan pada sistem ACS pada bagian modul *Underwriting* dan modul *Reasuransi* serta *deploy database environment* berdasarkan tugas yang diberikan oleh *kepala software tester*. Untuk pengujian yang dilakukan pada modul *Underwriting* dilakukan hasil analisa *issue Jira* dengan kode ACS-4268 pada *environment UAT*. Pada keterangan *Jira* menyatakan bahwa jangka waktu pada polis bagian objek

pertanggungan tidak tertampil dalam *field* berdasarkan pengujian *tester* lain dengan pengisian data menggunakan *upload* dokumen. Kemudian dilakukan analisa pada dengan melakukan template upload yang digunakan untuk melakukan *reproduce bug* dari *tester* yang mencoba *issue* tersebut.

Setelah dilakukan *reproduce* dengan template tersebut dan melakukan *upload* objek pertanggungan untuk pembuatan polis kecelakaan diri (TOC 1103



objectid	Kode_Tipe_Polis	Nama_Pertanggungan	No_Loan	Kode_Identitas	No_Identitas	Kode_Sex	Tempat_Lahir	Tgl_Lahir	Alamat
1	1103	Riswan	11111	KTP	3302230606560001	L	Jakarta	15/08/1991	-
2	1103	Tarsiti	22222	KTP	3302236120770001	P	Jakarta	21/02/1977	-

Gambar 3.29. Template upload yang digunakan untuk reproduce ACS-4268

PA plus PHK), maka dilakukan ekstrak respon dari *back-end* berupa JSON. Kemudian dilakukan pembuatan polis polis tanpa melakukan upload objek pertanggungan, lalu simpan polis tanpa dilakukan persetujuan berjenjang kemudian dilakukan pembukaan polis kembali, dan melakukan ekstrak respon dari *back-end* berupa JSON.

```

File Edit Selection View Go Run Terminal Help
{} json_data_policy_object_sample_TOC_sepada_motor.json query update json_data.sql

1
2  "usia": 27.0,
3  "paNdDateOfBirth": "15/08/1991",
4  "polisConditionText": "PA Plus PHK",
5  "noLoan": "11111",
6  "listObjAhliWaris": [
7    {
8      "heirName": "BRI CABANG PURWOKERTO",
9      "heirRelationship": "PAND_AW.ANAK_ANGKAT",
10     "heirBirthDate": "21/02/1978",
11     "heirAddress": "Beji Rt 004/001, Kec. Kedungbanteng",
12     "heirRelationshipLabel": "Anak Angkat"
13   }
14 ],
15 "paGenderText": "Laki-laki",
16 "tipeIndentitasDesc": "KTP",
17 "paDescription": "-",
18 "paNdPlaceOfBirth": "Jakarta",
19 "paNdAddress": "-",
20 "tipeIndentitasCode": "TIPE_IDENTITAS.1",
21 "paIssuredObject": "Riswan",
22 "paGenderCode": "JK.L",
23 "paAmtLol": 8.0E7,
24 "paNdNoIndentitas": "3302230606560001",
25 "txtSummaryRisiko": "-",
26 "paInformation": "di PHK",
27 "polisConditionCode": "A0686E7A-5F9A-4511-9C50-64696A90DFA3",
28 "paNdHariPertanggungan": 0.0,
29 "currencyCode": "IDR",
30 "productDescription": "1103 - PA Plus PHK"
31

```

Gambar 3.30. Respon *back-end* berupa JSON untuk pembuatan polis TOC 1103 dengan upload objek pertanggungan

```

{} json_data_policy_object_sample_TOC_sepada_motor.json query update json_data.sql {} Untitled-1 {} json_data_policy_object_sample.json X
D: > magang > pak Zein > UAT > ACS-4268 > {} json_data_policy_object_sample.json > ...
49  "paNdKurAmtHpnD": null,
50  "paNdKurPctRateND": null,
51  "paNdKurAmtPremiumND": null,
52  "paNdKurAmtDiscountND": null,
53  "paNdKurPctDiscountND": null,
54  "paNdKurAmtCommissionND": null,
55  "paNdKurPctCommissionND": null,
56  "paNdPlaceOfBirth": "Jakarta",
57  "paNdStartPeriod": "08/07/2019", //Start waktu objek pertanggungan PA Plus PHK
58  "paNdEndPeriod": "08/06/2020", // End waktu objek pertanggungan PA Plus PHK
59  "paNdName": null,
60  "paNdDateOfBirth": "08/07/2009",
61  "paNdAddress": "testing alamat jakarta ",
62  "paNdJob": null,
63  "tipeIndentitasCode": "TIPE_IDENTITAS.1",

```

Gambar 3.31. Respon *back-end* berupa JSON untuk pembuatan polis TOC 1103 tanpa *upload* objek pertanggungan

Berdasarkan hasil analisa kedua JSON tersebut, maka dapat dilihat tidak ada index array "*objectStartPeriode*" dan "*objectEndPeriode*" yang mencatat tanggal rentang jangka waktu polis tersebut pada JSON hasil upload objek pertanggungan, dapat dilihat dibandingkan dengan respon JSON polis yang tanpa upload objek

pertanggungan. Maka dapat disimpulkan data tersebut tidak tersimpan pada *database* dan tidak ada masalah terkait dari sistem ACS terhadap modul *underwriting*, namun kesalahan *template upload* yang digunakan. Hal ini dibuktikan dengan melakukan modifikasi data JSON pada tabel *UNDERWRITING.UDW_POLICY_OBJECT* yang memiliki keterangan *array* tersebut pada polis yang telah terbuat dengan *upload* objek pertanggungan.

```
D: > magang > pak Zein > UAT > ACS-4268 > query update json_data.sql
1  update UNDERWRITING.UDW_POLICY_OBJECT set JSON_DATA = '{
2      "usia": 27.0,
3      "paNdDateOfBirth": "15/08/1991",
4      "polisConditionText": "PA Plus PHK",
5      "noLoan": "11111",
6      "listObjAhliWaris": [
7          {
8              "heirName": "BRI CABANG PURWOKERTO",
9              "heirRelationship": "PAND_AW.ANAK_ANGKAT",
10             "heirBirthDate": "21/02/1978",
11             "heirAddress": "Beji Rt 004/001, Kec. Kedungbanteng",
12             "heirRelationshipLabel": "Anak Angkat"
13         }
14     ],
15     "paGenderText": "Laki-laki",
16     "tipeIndentitasDesc": "KTP",
17     "paDescription": "-",
18     "paNdPlaceOfBirth": "Jakarta",
19     "paNdAddress": "-",
20     "tipeIndentitasCode": "TIPE_IDENTITAS.1",
21     "paIssuredObject": "Riswan",
22     "paGenderCode": "JK.L",
23     "paAmtLo1": 8.0E7,
24     "paNdNoIdentitas": "3302230606560001",
25     "txtSummaryRisiko": "-",
26     "paInformation": "di PHK",
27     "paNdStartPeriod": "08/07/2019",
28     "paNdEndPeriod": "08/04/2020",
29     "polisConditionCode": "A0686E7A-5F9A-4511-9C50-64696A90DFA3",
30     "paNdHariPertanggungan": 0.0,
31     "currencyCode": "IDR",
32     "productDescription": "1103 - PA Plus PHK"
33 }' where POLICY_OBJECT_ID = '82B86201-394F-4FD0-8DD9-3A83FC26A028';
```

Gambar 3.32. Script query update untuk melakukan pengujian dari hasil analisa

Setelah melakukan pengujian dengan membuat *script update query* dengan memodifikasi JSON yang tersimpan pada tabel polis objek tersebut, maka dilakukan pengujian pada field jangka waktu polis tersebut dengan masuk ke

interface bagian *underwriting* untuk membuka polis tersebut. Setelah dibuka, maka terlihat *field* tersebut telah terisi terlihat pada gambar 3.33.

Gambar 3.33. Field objek jangka waktu yang terisi setelah melakukan modifikasi data pada database

Setelah menarik kesimpulan tersebut, maka dilakukan diskusi dengan *developer* terkait dengan *issue Jira* tersebut. Hasil diskusi tersebut, menyatakan bahwa kurangnya kolom pada *template upload* yang digunakan. Terdapat kekurangan kolom “Jkw_Pertanggungan_Start” dan “Jkw_Pertanggungan_End”. Kemudian dilakukan perbaikan template, dan dilakukan pengujian ulang terkait dengan polis TOC 1103 dengan upload objek pertanggungan.

objectid	Kode_Tipe_Polis	Jkw_Pertanggungan_Start	Jkw_Pertanggungan_End	Nama_Pertanggungan	No_Loan	Kode_Identitas	No_Identitas	Kode_Sex
1	1103	01/07/2019	01/07/2020	Riswan	11111	KTP	3302230606560001	L
2	1103	01/07/2019	01/07/2020	Tarsiti	22222	KTP	3302236120770001	P

Gambar 3.34. Dokumen *template upload* yang telah diperbaiki

Scenario ID		TSCEN-UDWR-001				Scenario Last Modified Date:			
Description		PROSES UNDERWRITING (Simple Approval - Level Kasie) - NORMAL CASE							
	N	Precondition	Test Procedure / Step	Validation Element	Expected Result	Status	Severity	Defect ID	Remarks
12	12		Lakukan analisa terhadap polis yang dipilih, lalu pada panel Disposisi input informasi hasil analisa, disposisi, saran, atau kesimpulan terhadap polis	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
19	13		Kirimkan polis ke Kasie untuk menjalankan approval polis	Database	Status polis berubah menjadi Approval Kasie	OK			
20	14	Login User Role : Kasie Cabang - Status Polis : Approval_KASIE_CAB	Buka dan tampilkan polis yang telah dibuat pada langkah sebelumnya	User Interface	Polis berhasil ditampilkan	FIX			tampilan belum sempurna
21	15		Bila polis tersebut memiliki ReAs, maka periksa nilai perhitungan pada tab Reasuransi Treaty apakah telah sesuai dengan skema treaty	User Interface	Nilai perhitungannya akan tergenerate otomatis sesuai dengan skema treaty	OK			
22	16		Lakukan analisa terhadap polis yang dipilih, lalu pada panel Disposisi input informasi hasil analisa, disposisi, saran, atau kesimpulan terhadap polis	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
23	17		Lakukan approval terhadap polis tersebut	Database	Status polis berubah menjadi Approved atau Setuju	OK			
24	18	Login User Role : Analis UDW Cabang - Status Polis : Approved/Setuju	Buka dan tampilkan polis yang telah di approved pada langkah sebelumnya	User Interface	Polis berhasil ditampilkan	FIX			tampilan belum sempurna
25	19		Lakukan proses penerbitan polis	Database	- Status polis berubah menjadi Terbit - Nomor Polis terbentuk	Fix			No. Polis running berdasarkan cabang dan Class bisnisnya
26	20		Cek jurnal akuntansi yang terbentuk	Integrasi Jurnal	Terbentuk jurnal akuntansi dg pola jurnal:	OK			

Master-Klaim

Report_2018_10_29

User Approval

TSCEN-UDWR-001

TSCEN-UDWR-002

Ts

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

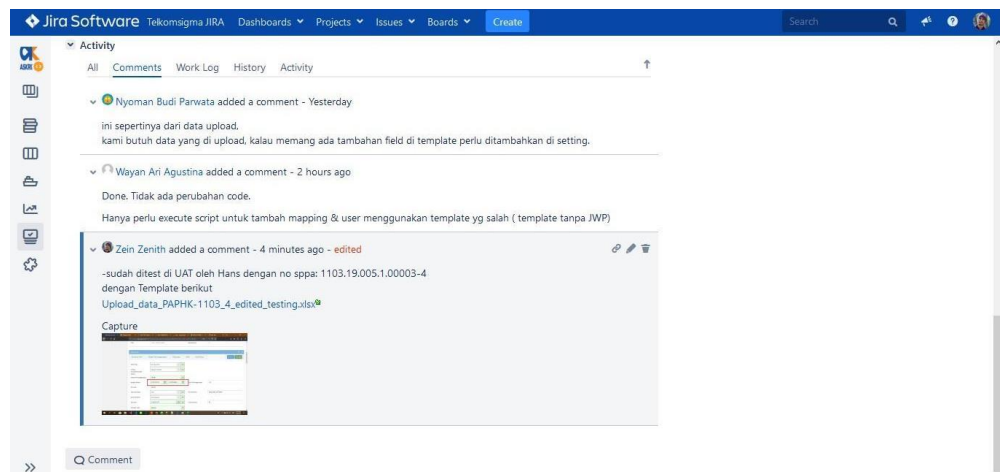
...

...

<

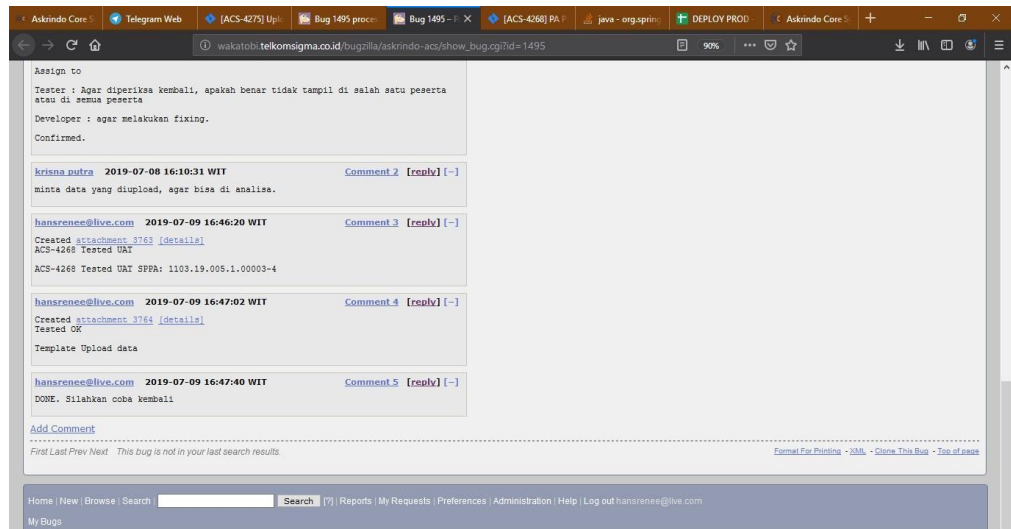
Gambar 3.35. Dokumen Test case yang digunakan untuk melakukan uji template upload pada pembuatan polis new

Setelah melakukan pengujian dengan upload template yang telah diperbaiki, maka *field* jangka waktu pertanggung jawaban objek pada polis tersebut terlihat. Sehingga menyimpulkan ini bukan sebuah *bug*. Hasil pengujian ini diperlihatkan pada kolom komentar dalam halaman *issue Jira* tersebut dan merubah status *issue Jira* tersebut menjadi *done* pada gambar 3.36.



Gambar 3.36. Pemberian komentar pada issue Jira terkait pengujian hasil pengujian tersebut

Setelah itu dilakukan juga perubahan status pada *bugzilla* yang terkait dengan *Jira* tersebut. Kode *bugzilla* yang terkait adalah 1495 pada gambar 3.37.



Gambar 3.37. Dilakukan *post* komentar terkait hasil pengujian *bug* terkait.

Lalu, dilakukan juga *deploy* pada migrasi dari sisi *database* sehingga diadakannya pengecekan dan eksekusi *query* yang dilakukan terhadap kode *Jira* terkait. Kode *Jira* tersebut adalah:

1. ACS-4262

Pada ACS-4262 dilakukan eksekusi *query* update untuk tabel master lookup untuk menambahkan nota premi dalam pencarian nota.

```
D: > magang > pak Zein > ACS SCRIPT > ACS-4262.sql
1  --ACS-4262
2  --revisi untuk Production
3
4  INSERT INTO [master].[M_LOOKUP]([LOOKUP_KEY], [GROUP_CODE], [KEY_ONLY], [LABEL], [IS_ACTIVE],
5  [VERSION], [CREATED_BY], [CREATED_DATE], [MODIFIED_BY], [MODIFIED_DATE])
6  VALUES('NOTE_TYPE_IS_MASTER.PREMIUM', 'NOTE_TYPE_IS_MASTER', 'Premium_Note', 'Premium Note', '1', 0,
7  'admin', '20181008 15:30:39.0', 'admin', '20181008 15:30:39.0')
8  GO
9
10 --select untuk violation memastikan
11 select [LOOKUP_KEY], [GROUP_CODE], [KEY_ONLY], [LABEL], [IS_ACTIVE],
12 [VERSION], [CREATED_BY], [CREATED_DATE], [MODIFIED_BY], [MODIFIED_DATE]
13 from [master].[M_LOOKUP] where LOOKUP_KEY = 'NOTE_TYPE_IS_MASTER.PREMIUM'
```

Gambar 3.38. Script *query* yang dibuat dan dicek pada ACS-4262

2. ACS-4290

Pada ACS-4290 dilakukan eksekusi *query insert* untuk menambah data pencarian laporan berdasarkan kode TOC pada tabel MASTER.M_LOOKUP_GROUP.

```
2  --ACS-4290
3
4  use ACS_DB_STAGING;
5
6  INSERT INTO MASTER.M_LOOKUP_GROUP
7  (GROUP_CODE, DESCRIPTION, IS_ACTIVE, VERSION, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE)
8  VALUES('TOC_REPORT', 'Toc Report', '1', 0, 'admin', '2018-11-26 15:38:14.930', NULL, NULL);
9
10
11  INSERT INTO MASTER.M_LOOKUP
12  (LOOKUP_KEY, GROUP_CODE, KEY_ONLY, LABEL, IS_ACTIVE, VERSION, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE)
13  VALUES('1102', 'TOC_REPORT', '1102', 'PSAKDI', '1', 0, 'admin', '2018-11-26 15:44:51.717', NULL, NULL);
14  INSERT INTO MASTER.M_LOOKUP
15  (LOOKUP_KEY, GROUP_CODE, KEY_ONLY, LABEL, IS_ACTIVE, VERSION, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE)
16  VALUES('1103', 'TOC_REPORT', '1103', 'PA Plus PHK', '1', 0, 'admin', '2018-11-26 15:44:51.717', NULL, NULL);
17  INSERT INTO MASTER.M_LOOKUP
18  (LOOKUP_KEY, GROUP_CODE, KEY_ONLY, LABEL, IS_ACTIVE, VERSION, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE)
19  VALUES('1104', 'TOC_REPORT', '1104', 'PA Aviation', '1', 0, 'admin', '2018-11-26 15:44:51.717', NULL, NULL);
20  INSERT INTO MASTER.M_LOOKUP
21  (LOOKUP_KEY, GROUP_CODE, KEY_ONLY, LABEL, IS_ACTIVE, VERSION, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE)
22  VALUES('1105', 'TOC_REPORT', '1105', 'PAND KUR', '1', 0, 'admin', '2018-11-26 15:44:51.717', NULL, NULL);
23
24
25  -- select 2x untuk memastikan
26  select LOOKUP_KEY, GROUP_CODE, KEY_ONLY, LABEL, IS_ACTIVE, VERSION, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE
27  FROM MASTER.M_LOOKUP where LOOKUP_KEY in('1102', '1103', '1104', '1105');
28
29  select GROUP_CODE, DESCRIPTION, IS_ACTIVE, VERSION, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE
30  from MASTER.M_LOOKUP_GROUP where GROUP_CODE = 'TOC_REPORT';
```

Gambar 3.39. Script query yang dibuat dan dicek pada ACS-4262

3. ACS-4283

Pada ACS-4283 terdapat *query developer* yang salah akan skema tabel tersebut untuk dijalankan. Maka dilakukan edit *query* sebelum kembali melakukan *execute query* tersebut. *Query* tersebut dilakukan untuk melakukan *update* label

pada eksekusi *query insert* untuk menambah data pencarian laporan berdasarkan kode TOC pada tabel MASTER.M_LOOKUP_GROUP

```
D: > magang > pak Zein > ACS SCRIPT > ACS-4283.sql
1  --ACS-4283
2  update master.M_LOOKUP set label = 'any one accident' where lookup_key = 'DEDUCTIBLE_UNIT.AOA'
3  update master.M_LOOKUP set label = 'any one carrying' where lookup_key = 'DEDUCTIBLE_UNIT.AOC'
4  update master.M_LOOKUP set label = 'any one location' where lookup_key = 'DEDUCTIBLE_UNIT.AOL'
5  update master.M_LOOKUP set label = 'any one motor' where lookup_key = 'DEDUCTIBLE_UNIT.AOM'
6  update master.M_LOOKUP set label = 'any one occurrence' where lookup_key = 'DEDUCTIBLE_UNIT.AOO'
7  update master.M_LOOKUP set label = 'any one person (Max. 4)' where lookup_key = 'DEDUCTIBLE_UNIT.AOP'
8  update master.M_LOOKUP set label = 'any one premises' where lookup_key = 'DEDUCTIBLE_UNIT.AOPR'
9  update master.M_LOOKUP set label = 'any one shipment' where lookup_key = 'DEDUCTIBLE_UNIT.AOS'
10 update master.M_LOOKUP set label = 'as charge' where lookup_key = 'DEDUCTIBLE_UNIT.ASCHARGE'
11 update master.M_LOOKUP set label = 'each and every loss' where lookup_key = 'DEDUCTIBLE_UNIT.EEL'
```

Gambar 3.40. Script query dari developer yang salah

```
D: > magang > pak Zein > ACS SCRIPT > ACS-4283.sql
1  --ACS-4283
2  update MASTER.M_LOOKUP set label = 'any one accident' where lookup_key = 'DEDUCTIBLE_UNIT.AOA'
3  update MASTER.M_LOOKUP set label = 'any one carrying' where lookup_key = 'DEDUCTIBLE_UNIT.AOC'
4  update MASTER.M_LOOKUP set label = 'any one location' where lookup_key = 'DEDUCTIBLE_UNIT.AOL'
5  update MASTER.M_LOOKUP set label = 'any one motor' where lookup_key = 'DEDUCTIBLE_UNIT.AOM'
6  update MASTER.M_LOOKUP set label = 'any one occurrence' where lookup_key = 'DEDUCTIBLE_UNIT.AOO'
7  update MASTER.M_LOOKUP set label = 'any one person (Max. 4)' where lookup_key = 'DEDUCTIBLE_UNIT.AOP'
8  update MASTER.M_LOOKUP set label = 'any one premises' where lookup_key = 'DEDUCTIBLE_UNIT.AOPR'
9  update MASTER.M_LOOKUP set label = 'any one shipment' where lookup_key = 'DEDUCTIBLE_UNIT.AOS'
10 update MASTER.M_LOOKUP set label = 'as charge' where lookup_key = 'DEDUCTIBLE_UNIT.ASCHARGE'
11 update MASTER.M_LOOKUP set label = 'each and every loss' where lookup_key = 'DEDUCTIBLE_UNIT.EEL'
```

Gambar 3.41. Script query yang telah dibetulkan dan disesuaikan terhadap database migrasi

4. ACS-4266

Pada ACS-4290 dilakukan eksekusi *query update* untuk mengubah tipe *field* Warranty menjadi *summernote* untuk polis TOC Harta Benda (TOC nomor 0101).

```

D: > magang > pak Zein > ACS SCRIPT > ACS-4266.sql
1  --ACS 4266
2
3  UPDATE ACS_DB_STAGING.MASTER.M_POLICY_MASTER_CONFIG set PARAM_TYPE='SUMMERNOTE',REQUIRED=0
4  WHERE PRODUCT_ID='0101' AND PARAM_CODE='WARRANTIES';
5
6
7
8  --query bawaan dari Jira
9  -- UPDATE ACS_DB_UAT.MASTER.M_POLICY_MASTER_CONFIG PARAM_TYPE='SUMMERNOTE',REQUIRED=0
10 -- WHERE PRODUCT_ID='0101' AND PARAM_CODE='WARRANTIES';
11
12

```

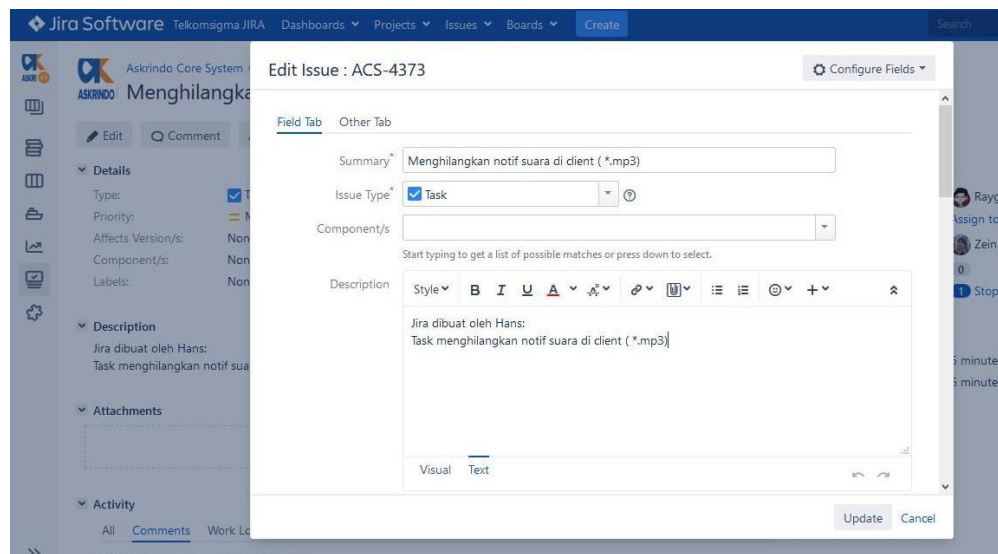
Gambar 3.42. Script query untuk ACS-4266

313	#046	Create System Parameter				
314	1	Buka menu Master, lalu pilih sub menu System Parameter	User Interface	Sistem akan menampilkan halaman System Parameter	OK	
315	2	Klik tombol + untuk membuat data System Parameter baru	User Interface	Sistem akan menampilkan form System Parameter	OK	
316	3a	Input semua informasi yang diperlukan terutama yang bersifat mandatory pada form System Parameter	Database	Sistem menampilkan halaman konfirmasi proses penambahan data (YES/ NO)	OK	
316		Lalu lakukan proses Simpan data				
317	3a1	Konfirmasi proses tambah data (YES)	Database	Sistem berhasil melakukan proses tambah data System Parameter ke Database	OK	
318	3a2	Konfirmasi proses tambah data (NO)	Database	Sistem batal melakukan proses tambah data System Parameter ke Database	OK	
318	3b	Input semua informasi yang diperlukan terutama yang bersifat mandatory pada form System Parameter	Database	Data System Parameter baru tidak berhasil ditambahkan ke Database	OK	
319		Lalu lakukan proses Pembatalan Simpan data				
320	#047	Update System Parameter				
321	1	Buka menu Master, lalu pilih sub menu System Parameter	User Interface	Sistem akan menampilkan halaman System Parameter	OK	
321	2	Pilih data System Parameter yang akan di update pada halaman inquiry dengan	User Interface	Sistem berhasil menampilkan satu atau beberapa	OK	

Gambar 3.43. Dokumen Test case yang digunakan untuk melakukan test summernote ACS-4290

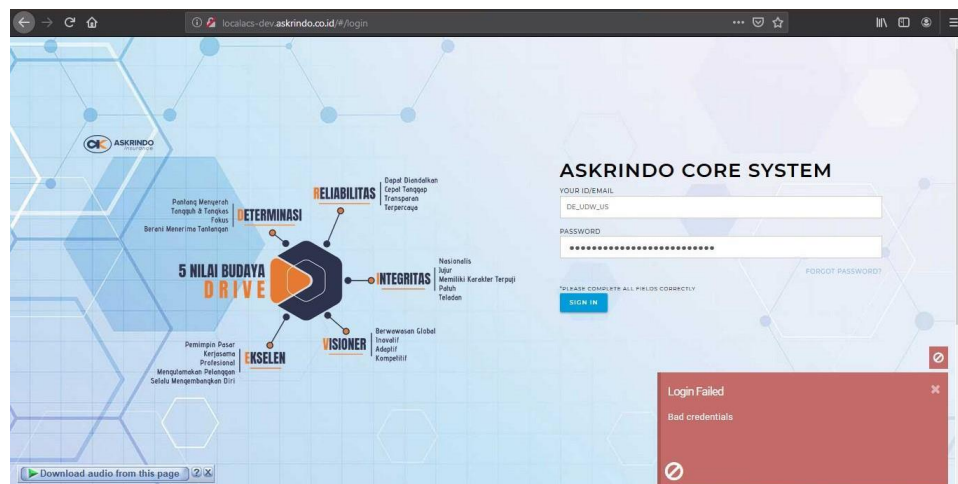
3.3.5 Minggu Lima, Testing, Deploy Execute Query Sesuai Jira, serta Pembuatan Task Jira

Pada minggu kelima, dilakukan pengujian lanjutan pada sistem ACS serta melakukan *deploy* dan *execute* terhadap *database environment staging* serta melakukan pembuatan *issue Jira* dengan tipe *task* untuk *developer* dan dilakukan pengujian atas penambahan fitur tersebut dalam *staging* dan migrasi yang ditugaskan dan dilakukan *deploy* oleh kepala *software tester*.



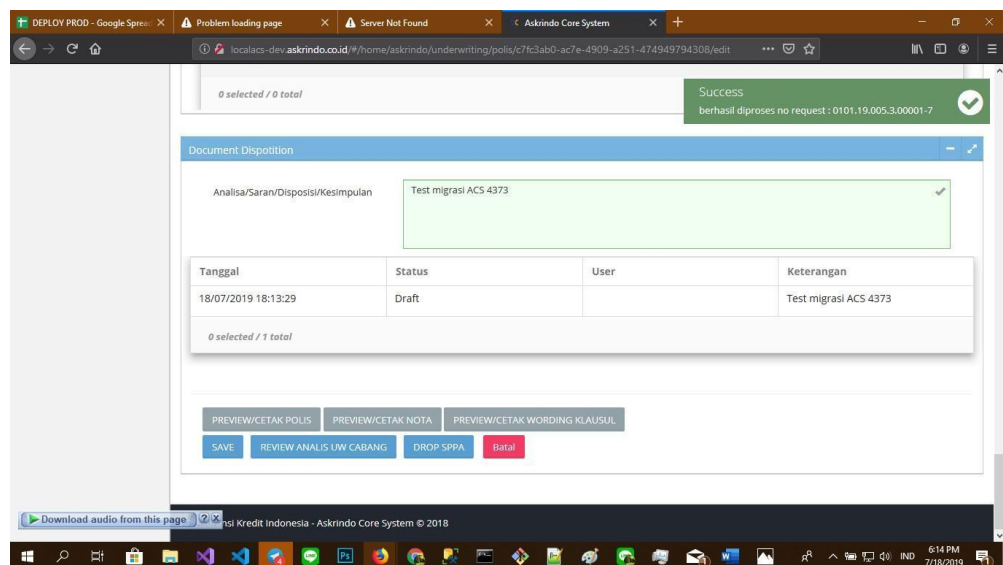
Gambar 3.44. Task Jira ACS-4373 yang telah dibuat

Jira pada gambar 3.44, dibuat atas permintaan pihak perusahaan Askrindo terkait dengan hal penambahan suara setiap notifikasi yang muncul pada sistem ACS.



Gambar 3.45. Pengujian suara notifikasi sewaktu login sistem ACS pada migrasi

Setelah dilakukan pengujian terhadap notifikasi tersebut berhasil terdengar. Lalu hasil tersebut dikuatkan kembali terhadap notifikasi dari *download manager plugin* pada *browser* yang sekarang terlihat pada pojok kiri bawah terdapat adanya notifikasi untuk melakukan unduh suara dalam halaman web tersebut.



Gambar 3.46. Pengujian suara notifikasi sukses pada pembuatan polis

Kemudian dilanjutkan dengan melakukan *deploy* pada *environment staging* serta migrasi dengan melakukan pembuatan *query* berdasarkan diskusi dengan *developer* yang terkait dengan kode *Jira* ACS-3931. *Jira* tersebut melakukan pembuatan tabel baru serta menjadi tabel backup terhadap polis berdasarkan ketentuan migrasi pada sebelumnya yang telah disepakati oleh perusahaan Askrindo.



```
DDL_UDW_POLICY_MIG(UAT)REI_v2.sql x whatsapp.php ●
D: > magang > pak Zein > ACS SCRIPT > DDL_UDW_POLICY_MIG(UAT)REI_v2.sql
1 use ACS_DB;
2
3 select * into REINSURANCE.UDW_POLICY_MIG
4 from UNDERWRITING.UDW_POLICY;
5
6 select case TABLE_CATALOG
7     when 'ACS_DB_UAT' then 'Sesuai'
8     else 'Salah'
9     END
10 as 'Posisi DB', TABLE_CATALOG as 'Database/Skema', TABLE_SCHEMA, TABLE_NAME, DATA_TYPE
11 from INFORMATION_SCHEMA.COLUMNS where TABLE_SCHEMA = 'REINSURANCE'
12 and TABLE_NAME = 'UDW_POLICY_MIG';
13
14 select * from INFORMATION_SCHEMA.TABLES where TABLE_SCHEMA = 'REINSURANCE'
15 and TABLE_NAME = 'UDW_POLICY_MIG';
16
17
```

Gambar 3.47. Query ACS-3931 yang dibuat dan untuk dilakukan *deploy* pada *environment staging* dan migrasi

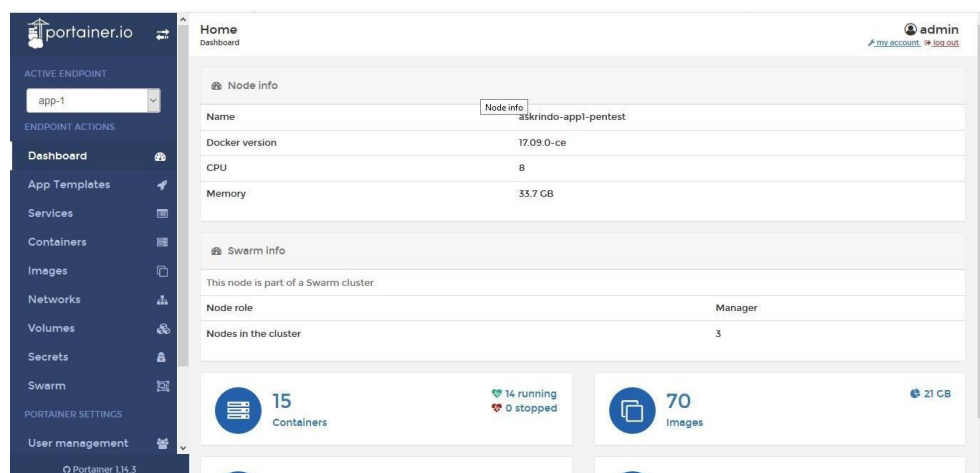
Atas diskusi tersebut, dilakukan pembuatan *script query* untuk membuat tabel *UNDERWRITING.UDW_POLICY* dengan struktur yang sudah ada pada tabel polis sebelumnya.

3.3.6 Minggu Enam, Pengaturan *Log Server*, *extract log server*, *NFT Testing*, dan *deploy Execute Query* sesuai *Jira*

Pada minggu keenam, dilakukan pembelajaran terhadap pengaturan *log* pada server dari setiap *environment*, melakukan ekstrak *log* untuk dianalisa, kemudian melakukan *Non Functional Testing* terhadap *server* dan dilakukan pengujian serta *execute query* untuk melakukan *deploy* terkait *Jira* yang diberikan oleh kepala *software tester*.

Pada minggu keenam, dilakukan pembelajaran terhadap pengaturan *log* pada server dari setiap *environment* serta dilakukan ekstrak *log* dari setiap *service*

dari suatu server untuk dianalisa. Untuk melakukan ekstrak *log* dari setiap *services* maka digunakan *tools portainer*. Menurut dokumentasi dari *tools* tersebut, *portainer* merupakan alat yang berfungsi untuk mengatur manajemen *container* dari dalam sebuah *server* yang terdiri dari beberapa *services*.



Gambar 3.48. Tampilan *dashboard* pada *portainer* dalam *server environment staging*

Seperti lingkungan pengembangan, *portainer* juga memiliki pengaturan secara terpisah. Kemudian kepala *tester* memberi tugas untuk melakukan *Non Functional Test* yang merupakan pengujian terhadap hal yang mempengaruhi performa dari sistem ACS tersebut dengan kode *Jira ACS-4311* terhadap level *log* dari sistem. Dalam *framework spring boot* yang digunakan dalam sistem ACS ini, terdapat beberapa level yang diantaranya adalah:

1. *Error*

Kejadian yang ditampilkan pada *log* hanya berupa kondisi yang tak terkendali dan diluar dugaan yang diprogram dan menampilkan jejak serta *kode error*.

2. *Info*

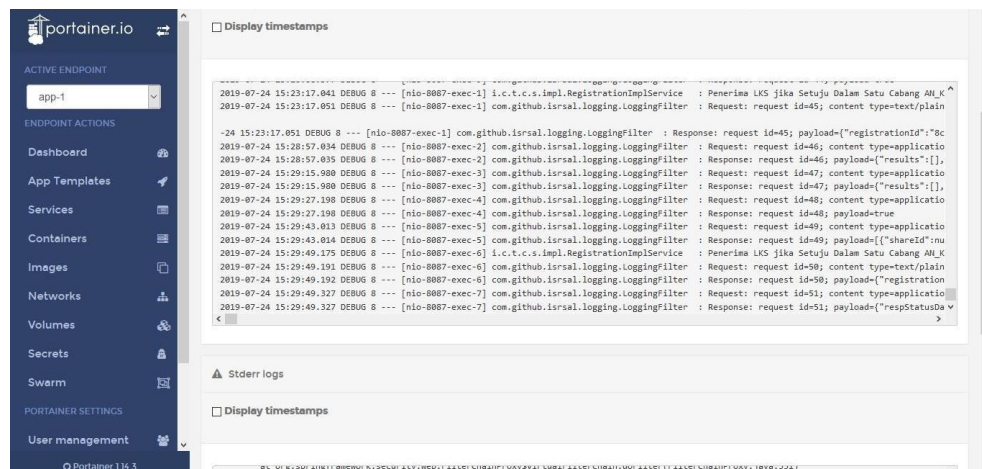
Kejadian yang ditampilkan hanya sekedar dari *event* atau kejadian terhadap eksekusi sebuah *code* tertentu dan tidak menampilkan *kode error* apabila terjadi sebuah malfungsi atau error, hanya indikasi mulai dan berhenti.

3. *Debug*

Kejadian yang ditampilkan berupa setiap event atau kejadian eksekusi dari sistem mulai sebuah *code* dari awal sampai akhir dan melakukan pencatatan *kode error* tersebut.

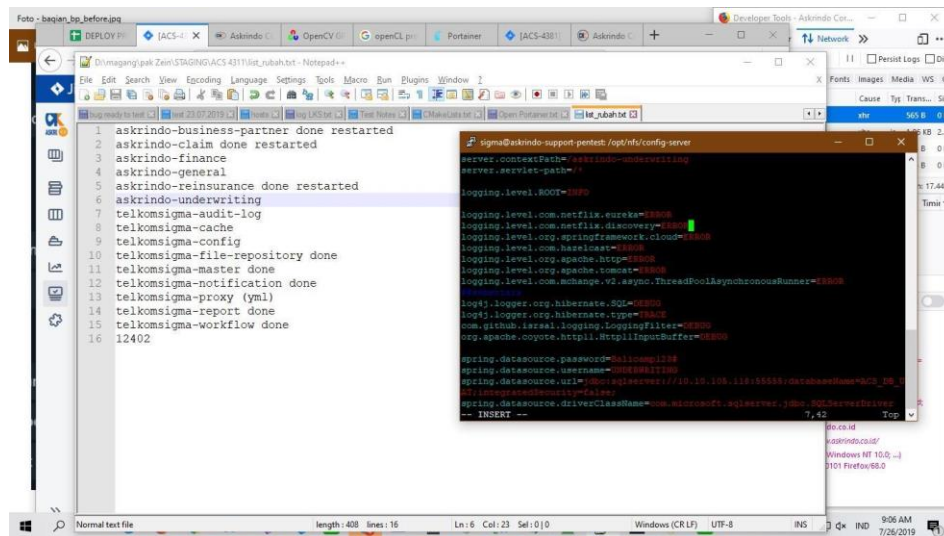
4. *Trace*

Tingkatan *log* paling *detail* dalam menampilkan informasi, dimana setiap kejadian sampai awal hingga akhir ditampilkan seperti level *log debug* sampai menuliskan *code* seperti fungsi yang dijalankan.



Gambar 3.49. Log yang dihasilkan pada *server staging* pada *service klaim*

Berdasarkan diskusi dengan *developer* terkait *Jira* tersebut, maka dilakukan perubahan level *log* menjadi *info* terhadap semua *services* yang berjalan agar menghemat *file log*, serta performa sistem yang sedang berjalan meningkat. Dilakukan *login* menggunakan koneksi protokol SSH (*Secure Shell*) ke dalam *server staging* dengan membuka file berformat ekstensi *.properties*



Gambar 3.50. Perubahan file log pada server staging ingin dilakukan deploy staging pada service underwriting

Kemudian dilakukan juga pengujian pada *staging* dan migrasi atas *bug* yang telah ditemukan oleh pihak pengguna sistem ACS dalam *production* melalui bugzilla dengan kode 1567. *Bug* tersebut dalam proses pembuatan polis terdapat objek pertanggungans tidak tertampil secara berurutan, sehingga dalam pencetakan *preview* polis terjadi kesalahan pada *TOC Document in Safe* (kode TOC 2003).

Halaman ini adalah bagian dari SPPA No. 2003.19.005.3.00001-7
This page is a forming part of SPPA No. 2003.19.005.3.00001-7

Asuransi Cash In Safe (Dokumen) a/n PURWANTO
Deklarasi Tanggal 01 Agustus 2019 s/d 31 August 2019

No	Nama Debitur	No CIF No Pinjaman	Periode Pertanggungan		Jenis Dokumen Yang Diasuransikan	Nilai Pertanggungan		Premi
			Awal	Akhir				
1	Testing nama deb 2	pinjaman 2	07/08/2019	07/08/2020	rahasia 2	IDR	20,000,000.00	IDR 300,000.00
2	DEBITUR 3	PINJ 3	07/08/2019	07/08/2020	Rahasia 3	IDR	15,000,000.00	IDR 165,000.00
3	deb4	pinj4	07/08/2019	07/08/2020	Rahasia 4	IDR	25,000,000.00	IDR 375,000.00
4	Testing Debitur 5	pinj 5	07/08/2019	07/08/2020	Rahasia 5	IDR	30,000,000.00	IDR 450,000.00
5	Testing deb 6	Test pinj6	07/08/2019	07/08/2020	Rahasia 6	IDR	37,000,000.00	IDR 458,800.00
6	Testi deb 1	test pinj 1	07/08/2019	07/08/2020	raHASIA 1	IDR	50,000,000.00	IDR 550,000.00
Jumlah						IDR	177,000,000.00	IDR 2,298,800.00

Semarang, 07 August 2019

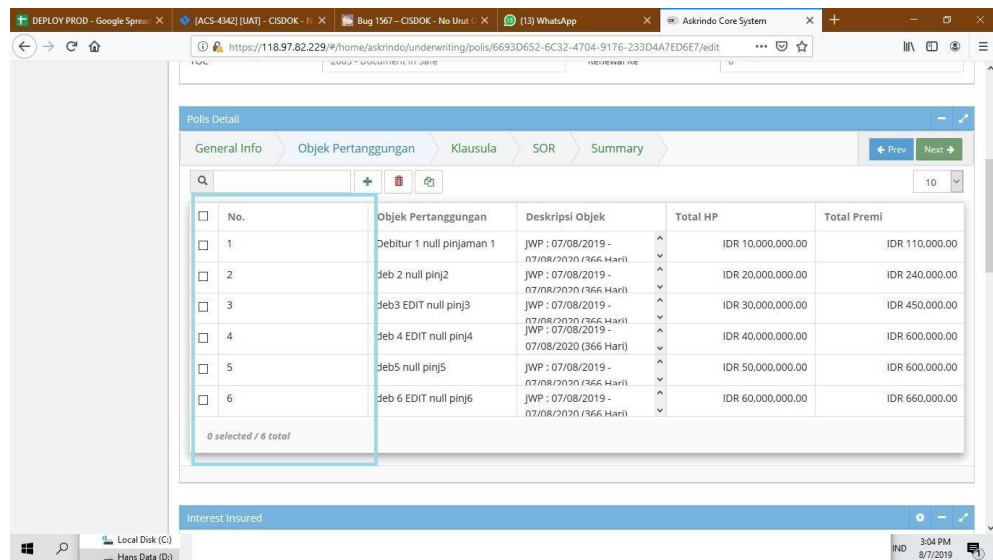
Gambar 3.51. Gambar *preview* polis terkait objek yang ditanggungan tidak tertampil urut

Askrindo Core System - Test Scenario									
1	Back To Cover								
4	Scenario ID	TSCEN-UDWR-001							
5	Description	PROSES UNDERWRITING (Simple Approval - Level Kasie) - NORMAL CASE							
6	Precondition								
7	0003 Akseptasi Simpel Polis								
8	1 Login User Role : Front Desk	Buka menu Akseptasi untuk membuat Polis baru	User Interface	Sistem menampilkan form pembuatan polis baru	OK				
9	2	Input semua informasi yang diperlukan terutama yang bersifat mandatory pada tab General Info	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK				
10	3	Tambahkan Object pada tab Object Pertanggungan A. Input semua informasi object pertanggungan yang diperlukan, terutama yang bersifat mandatory B. Pada panel Risiko tekan tombol Hitung Premi untuk menghitung premi yang memiliki tipe Basic Cover (Mandatory) / tekan tombol "+" untuk menambah risiko yang memiliki tipe Perluasan (Optional) C. Simpan informasi pada tab Object Pertanggungan	User Interface	- Hasil premi sesuai dengan metode perhitungan yang digunakan - Semua informasi yang terinput berhasil tersimpan sementara	OK				
11	4	Pada tab Klausula pilih Klausula dan Warranty	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK				
12	5	Jalankan fungsi hitung premi pada tab Summary	User Interface	Sistem akan melakukan perhitungan premi dan akan menampilkan ringkasan premi dan rincian nota	OK				
13	6	Pada panel Disposisi, input analisa atau disposisi terhadap polis tersebut	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK W/ NOTE				Cara pencarian dan cara penginputan Profil ID tertanggung. Cara pencarian dan penginputan Blok risiko masih belum sesuai

Gambar 3.52. Dokumen *Test case* yang digunakan untuk melakukan pengujian nomor urut objek pertanggungan

Setelah dilakukan diskusi dengan *developer* terkait, maka dibuatkan *issue bug* dengan kode *Jira ACS-4342*. Setelah *developer* melakukan *fix* terhadap *bug*

tersebut maka dilakukan pengujian terkait dengan kode *Jira* tersebut pada *environment staging*.



No.	Objek Pertanggungan	Deskripsi Objek	Total HP	Total Premi
1	Debitur 1 null pinjaman 1	JWP : 07/08/2019 - 07/08/2020 (366 Hari)	IDR 10,000,000.00	IDR 110,000.00
2	deb 2 null pinj2	JWP : 07/08/2019 - 07/08/2020 (366 Hari)	IDR 20,000,000.00	IDR 240,000.00
3	deb3 EDIT null pinj3	JWP : 07/08/2019 - 07/08/2020 (366 Hari)	IDR 30,000,000.00	IDR 450,000.00
4	deb 4 EDIT null pinj4	JWP : 07/08/2019 - 07/08/2020 (366 Hari)	IDR 40,000,000.00	IDR 600,000.00
5	deb5 null pinj5	JWP : 07/08/2019 - 07/08/2020 (366 Hari)	IDR 50,000,000.00	IDR 600,000.00
6	deb 6 EDIT null pinj6	JWP : 07/08/2019 - 07/08/2020 (366 Hari)	IDR 60,000,000.00	IDR 660,000.00

Gambar 3.53. Pengujian ACS-4342 pada *environment staging*

Setelah proses *deploy* selesai oleh kepala *software tester*, pengujian pada *environment staging* terkait ACS-4342 dilakukan. Hasil dari pengujian tersebut, *bug* telah diperbaiki, baik dalam penambahan objek secara manual maupun menggunakan *upload* dokumen objek pertanggungan.

Halaman ini adalah bagian dari No SPPA 2003.19.005.3.00003-4

Asuransi Cash In Safe (Dokumen) a/n SULAR

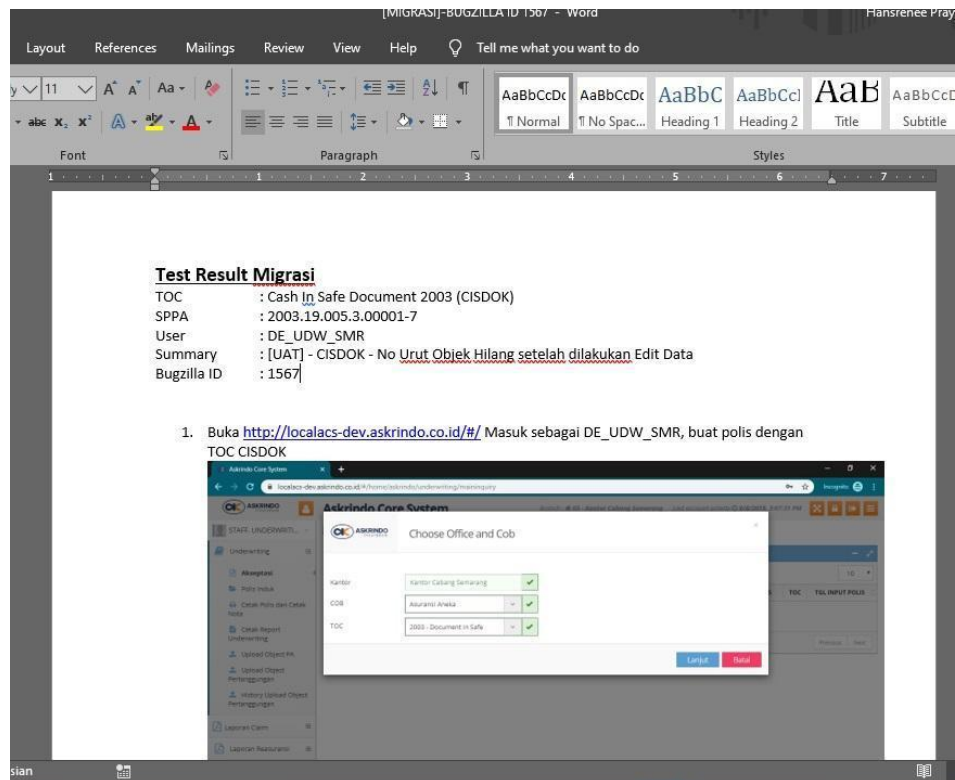
Deklarasi Tanggal 01 Agustus 2019 s/d 31 August 2019

No	Nama Debitur	No CIF No Pinjaman	Periode Pertanggungan		Jenis Dokumen Yang Diasuransikan	Nilai Pertanggungan	Premi
			Awal	Akhir			
1	Debitur 1	pinjaman 1	07/08/2019	07/08/2020	rahasia 1	IDR 10,000,000.00	IDR 110,000.00
2	deb 2	pinj2	07/08/2019	07/08/2020	rahasia 2	IDR 20,000,000.00	IDR 240,000.00
3	deb3 EDIT	pinj3	07/08/2019	07/08/2020	rahasia 3	IDR 30,000,000.00	IDR 450,000.00
4	deb 4 EDIT	pinj4	07/08/2019	07/08/2020	rahasia 4	IDR 40,000,000.00	IDR 600,000.00
5	deb5	pinj5	07/08/2019	07/08/2020	rahasia 5	IDR 50,000,000.00	IDR 600,000.00
6	deb 6 EDIT	pinj6	07/08/2019	07/08/2020	rahasia 6	IDR 60,000,000.00	IDR 660,000.00
Jumlah						IDR 210,000,000.00	IDR 2,660,000.00

Semarang, 07 Agustus 2019

Gambar 3.54. Preview polis setelah dilakukan perbaikan pada *environment staging*

Kemudian dilakukan pengujian ACS-4342 di *environment* migrasi, sesuai aturan apabila *staging* sudah selesai dilakukan pengujian, maka akan dilakukan *deploy* serta dilakukan pengujian serupa pada *migrasi* serta dilakukan pembuatan dokumentasi pengujian yang berupa *test result* dalam *environment* migrasi.



Gambar 3.55. Pembuatan dokumen *test result* dari hasil pengujian pada *environment* migrasi

Setelah membuat dokumentasi tersebut, dilampirkan pada hasil tersebut di dalam *bugzilla* dan dilakukan perubahan status *bug* tersebut menjadi *resolved*.

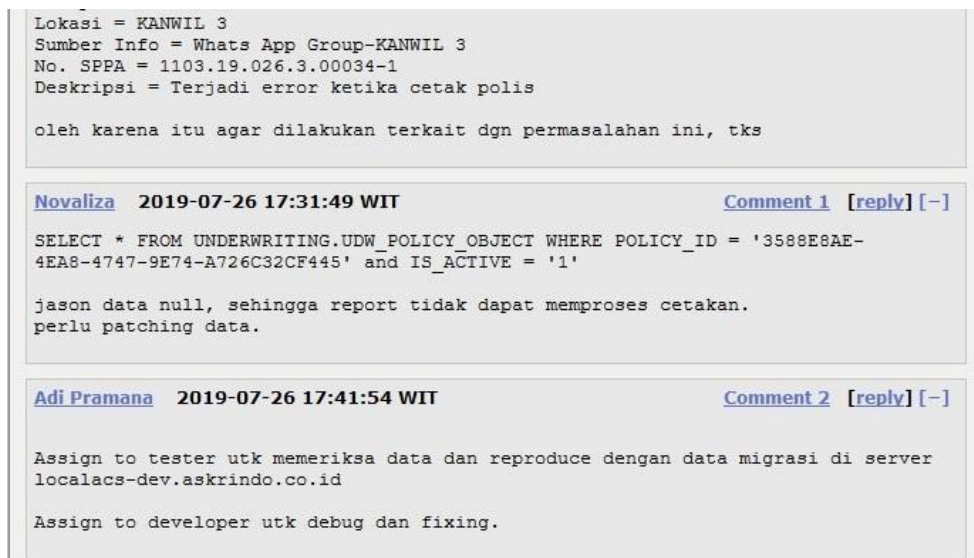
3.3.7 Minggu Tujuh, Pengecekan dan *Reproduce bugzilla* serta *deploy Execute Query* sesuai *Jira*

Pada minggu ketujuh, dilakukan *reproduce bug* yang dilaporkan oleh pihak pengguna sistem ACS pada *environment production* dengan kode bugzilla 1643. Berdasarkan pada *bugzilla 1643*, dilaporkan bahwa terjadi error ketika melakukan cetak polis. Setelah dilakukan diskusi oleh developer secara teknis, kolom JSON

pada tabel polis tersebut memiliki nilai *null* atau kosong, sehingga harus dilakukan *patching* data atau perubahan data secara “*hard code*” pada *database*.

```
ACS-3423.sql reproduce_cek.sql {} result_json_data_reproduce.code-workspace
D: > magang > pak Zein > MIGRASI > Reproduce > bugzilla_1643 > {} json_data_prod.JSON > ...
1
2 "txtSummaryRisiko": null,
3 "registerNo": null,
4 "polisConditionCode": null,
5 "additionalInformation": null,
6 "polisConditionText": null,
7 "itemCode": null,
8 "paIssuredObject": null,
9 "paDescription": null,
10 "currencyCode": null,
11 "paAmtLol": null,
12 "paPctRate": null,
13 "paAmtPremium": null,
14 "paAmtDiscount": null,
15 "paPctDiscount": null,
16 "paAmtCommission": null,
17 "paPctCommission": null,
18 "paDeductable": null,
19 "paInformation": null,
20 "productDescription": null,
21 "paGenderCode": null,
22 "paGenderText": null,
23 "noLoan": null,
24 "paNdKurTypeCode": null,
25 "paNdKurTypeText": null,
26 "paNdKurHpConditionCode": null,
27 "paNdKurHpConditionText": null,
28 "paNdKurIdTypeCode": null,
29 "paNdKurIdTypeText": null,
30 "paNdKurIdDesc": null,
31 "paNdKurGenderCode": null,
32 "paNdKurGenderText": null,
33 "paNdKurHealthCondition": null.
```

Gambar 3.56. Data JSON yang diambil dari *database production* oleh *developer* terkait



Gambar 3.57. Komentar diskusi dalam halaman *bugzilla* kode 1643

Berdasarkan, komentar langsung dari *business analyst* yang merupakan pak Adi Pramana pada gambar 3.57, maka dilakukan *reproduce bug* tersebut pada *environment* migrasi. Setelah dilakukan *reproduce* terhadap *bugzilla* 1643 yang merupakan proses pembuatan polis TOC 1103, polis tersebut berhasil disimpan dan dicetak.

ORIGINAL

IKHTISAR POLIS

ASURANSI KECELAKAAN DIRI PLUS

1. <u>No SPPA</u>	:	1103.19.005.3.00001-7	New
SPPA No	:		
2. <u>Nama Pemegang Polis</u>	:	KOPERASI PEGAWAI NEGERI ANGGARAN	
Policyholder Name	:		
3. <u>Alamat Pemegang Polis</u>	:	Gedung Keuangan Negara Yogyakarta Jalan Kusumanegara No. 11	
Policyholder Address	:	Kec. Umbulhajo, YogyakartaINDONESIA	
4. <u>Nama Tertanggung</u>	:	KOPERASI PEGAWAI NEGERI ANGGARAN	
Insured Name	:		
5. <u>No. Loan</u>	:	08170000000000	
Loan No.	:		
6. <u>No. Identitas (KTP)</u>	:	36709423049324329483294829	
Identity No. (KTP)	:		
7. <u>Tempat / Tanggal Lahir</u>	:	Ambarawa / 30 July 1990	
Place / Date of Birth	:		
8. <u>Alamat Tertanggung</u>	:	Testing Alamat dimana2	
Insured Address	:		
9. <u>Ahli Waris</u>	:		
Heir	:		
No.	Nama	Hubungan dengan Tertanggung	

Gambar 3.58. Polis yang berhasil dicetak dalam melakukan *reproduce bug* 1643 pada migrasi

Kemudian dilakukan juga pembuatan query untuk melakukan pengecekan pada kolom JSON_DATA yang berada pada tabel polis pada server migrasi. Setelah dilakukan pengecekan, nilai data JSON tersebut tidak *null*, dan menyimpan informasi polis dengan sesuai.

```
D: > magang > pak Zein > MIGRASI > Reproduce > bugzilla_1643 > reproduce_cek.sql
1  --query cek Reproduce
2  select REQUEST_NO, UNDERWRITING.UDW_POLICY.POLICY_ID, JSON_DATA from UNDERWRITING.UDW_POLICY join
3  UNDERWRITING.UDW_POLICY_OBJECT on UNDERWRITING.UDW_POLICY_OBJECT.POLICY_ID = UNDERWRITING.UDW_POLICY.POLICY_ID
4  where REQUEST_NO = '1103.19.005.3.00001-7' AND UNDERWRITING.UDW_POLICY.IS_ACTIVE = 1;
5
```

Gambar 3.59. Script query yang dibuat untuk melakukan pengecekan data JSON pada database migrasi

```
ek.sql  {} result_json_data_reproduce.code-workspace  DDL_UDW_POLICY_MIG(UAT)REI_v2.sql
D: > magang > pak Zein > MIGRASI > Reproduce > bugzilla_1643 > {} result_json_data_reproduce.JSON >
1  [
2      "txtSummaryRisiko": "",
3      "registerNo": null,
4      "polisConditionCode": "A0686E7A-5F9A-4511-9C50-64696A90DFA3",
5      "additionalInformation": null,
6      "polisConditionText": "PA Plus PHK",
7      "itemCode": null,
8      "paIssuredObject": "Testing bugzilla 1643 Jaminan",
9      "paDescription": "Testsing Desc ",
10     "currencyCode": "IDR",
11     "paAmtLol": 700000000,
12     "paPctRate": null,
13     "paAmtPremium": 0,
14     "paAmtDiscount": 0,
15     "paPctDiscount": 0,
16     "paAmtCommission": 0,
17     "paPctCommission": 0,
18     "paDeductable": null,
19     "paInformation": null,
20     "productDescription": "1103 - PA Plus PHK",
21     "paGenderCode": "JK.L",
22     "paGenderText": "Laki-laki",
23     "noLoan": "08170000000000",
24     "paNdKurTypeCode": null,
25     "paNdKurTypeText": null,
26     "paNdKurHpConditionCode": null,
27     "paNdKurHpConditionText": null,
```

Gambar 3.60. Data JSON yang diambil dari database migrasi dalam reproduce bug

Lalu dilakukan perubahan status *bug* yang ada didalam *bugzilla* menjadi *resolved* dan melampirkan hasil polis yang telah tercetak dengan melampirkan file dengan ekstensi PDF pada halaman diskusi *bug* 1643.

First Last Prev Next This bug is not in your last search results.

Bug 1643 - Terjadi error ketika cetak polis (edit) [Save Changes](#)

Status: RESOLVED WORKSFORME ([edit](#))

Product:

Component:

Version:

Hardware:

Importance:

Assigned To: [Adi Pramana \(edit\) \(take\)](#)

URL:

Keywords:

Tags:

Depends on:

Blocks:

Show dependency [tree](#) / [graph](#)

Reported: 2019-07-26 16:10 WIT by [Widyanahar](#)

Modified: 2019-07-30 10:04 WIT ([history](#))

CC List: 3 users including you ([edit](#))

See Also: ([add](#))

Issue Relation:

TESTING STEPS:

JIRA ID:

Revision ID:

Scenario ID:

Scenario Step ID:

CR Form Number:

CR Mandays:

Flags: None yet set ([set flags](#))

Orig. Est.	Current Est.	Hours Worked	Hours Left	%Complete	Gain	Deadline
0.0	0.0	0.0 + 0	0.0	0	0.0	<input type="text"/>

[Summarize time \(including time for bugs blocking this bug\)](#)

Gambar 3.61. Perubahan status *bugzilla* 1643

Lokasi = KANWIL 3
Sumber Info = Whats App Group-KANWIL 3
No. SPPA = 1103.19.026.3.00034-1
Deskripsi = Terjadi error ketika cetak polis
oleh karena itu agar dilakukan terkait dgn permasalahan ini, tks

[Novaliza](#) 2019-07-26 17:31:49 WIT [Comment 1](#) [reply](#) [-]

SELECT * FROM UNDERWRITING.UDW POLICY OBJECT WHERE POLICY_ID = '3588E8AE-4EA8-4747-9E74-A726C32CF445' and IS_ACTIVE = '1'

jason data null, sehingga report tidak dapat memproses cetakan.
perlu patching data.

[Adi Pramana](#) 2019-07-26 17:41:54 WIT [Comment 2](#) [reply](#) [-]

Assign to tester utk memeriksa data dan reproduce dengan data migrasi di server localacs-dev.askrindo.co.id

Assign to developer utk debug dan fixing.

[hansrenee@live.com](#) 2019-07-30 10:02:26 WIT [Comment 3](#) [reply](#) [-]

Created [attachment 4038](#) [\[details\]](#)
Hasil laporan reproduce di Migrasi

[hansrenee@live.com](#) 2019-07-30 10:04:46 WIT [Comment 4](#) [reply](#) [-]

Untuk cetak laporan sudah di reproduce di Migrasi berhasil cetak

Keterangan:
-No SPPA: 1103.19.005.3.00001-7
-Username 1092 atas nama Paramita Nurinaputri

Gambar 3.62. Melakukan respon pada halaman *bugzilla* 1643

Description		PROSES UNDERWRITING (Simple Approval - Level Kasie) - NORMAL CASE							
6	W	Precondition	Test Procedure / Step	Validation Element	Expected Result	Status	Severity	Defect ID	Remarks
29		1004 Akseptasi Simple Polis dengan Premi Full							
29	1	Login User Role: Front Desk	Buka menu Akseptasi untuk membuat Polis baru	User Interface	Sistem menampilkan form pembuatan polis baru	OK			
30	2		Input semua informasi yang diperlukan terutama yang bersifat mandatory pada tab General Info	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
31	3		Pada tab General Info pilih Premi Full untuk metode perhitungan polis	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
32	4		Tambahkan Object pada tab Object Pertanggungan A. Input semua informasi object pertanggungan yang diperlukan, terutama yang bersifat mandatory B. Pada panel Risiko tekan tombol Hitung Premi untuk menghitung premi yang memiliki tipe Basic Cover (Mandatory) / tekan tombol "+" untuk menambah risiko yang memiliki tipe Perluasan (Optional). C. Simpan informasi pada tab Obyek Pertanggungan	User Interface	Semua informasi yang terinput berhasil tersimpan sementara - Nilai premi sesuai dengan metode perhitungan full premi - Premi Full = Harga Pertanggungan x Rate	OK			
33	5		Pada tab Klausula pilih Klausula dan Warranty	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
34	6		Jalankan fungsi hitung premi pada tab Summary	User Interface	Sistem akan melakukan perhitungan premi dan akan menampilkan ringkasan dan nota premi	OK			
35	7		Pada panel Disposisi, input analisa atau disposisi terhadap polis tersebut	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
	8		Kirimkan polis ke Analis Underwriting untuk menjalankan workflow persetujuan polis	Database	Semua input yang dilakukan dari step no 1 s/d 7 berhasil tersimpan di database - No Reo SPAA terbentuk sesuai	OK			

Gambar 3.63. Dokumen *Test case* yang digunakan untuk melakukan reproduce *bugzilla* 1643

Selanjutnya, dilakukan pengujian terhadap *bug* ACS-4410. *Issue* ini dibuat karena terdapat *bug* pada field kantor cabang penerbit polis dan kantor cabang pengajuan klaim tidak sesuai. Maka dilakukan diskusi dengan *developer* terkait dengan modul klaim pada *Jira* tersebut untuk melakukan perbaikan. Dilakukan pembuatan polis TOC *Oil and Gas* (kode TOC 0801) dalam pengujian tersebut pada bagian *underwriting* cabang Padang dan dilakukan proses klaim pada cabang Semarang.

Registrasi Klaim

ASKRINDO Insurance

Pencarian Polis

Kantor Cabang: Kantor Cabang Padang x v

COB: Pilih COB v

TOC: Pilih TOC v

Nomor Polis: 0801.19.020.3.00001-4/00

Nomor Sertifikat: Nomor Sertifikat

Nama Tertanggung: Nama Tertanggung

Reset Cari

10 v

Aksi	Nomor Polis	Nomor Sertifikat	Nama Tertanggung	Cabang	Produk
<input checked="" type="radio"/>	0801.19.020.3.00001-		HASTARI	Wilayah VI	Oil & Gas

1 total

Gambar 3.64. Proses pencarian polis yang telah dibuat untuk dilakukan klaim pada kantor cabang Semarang

Pada gambar 3.57, terdapat polis yang telah terdaftar sesuai cabang. Kemudian polis tersebut dipilih dan dalam pengujian ini, *bug* telah teratasi pada *environment* migrasi.

Askrindo Core System

Branch : 05 - Kantor Cabang Semarang Last account activity 7/29/2019, 10:53:10 AM

Home / Claim / Registrasi Klaim Editor

Forms > Detail Registrasi Klaim

Registrasi Klaim

No Registrasi Klaim: K.0801.19.005.3.00002 Tanggal Entri: 2019-07-29 17:52:36.997

Status Klaim: Registrasi User: AN_KLAIM_SMR

Cabang Penerbit Polis: Kantor Cabang Padang Kantor Cabang: Kantor Cabang Semarang

Gambar 3.65. Field cabang penerbit polis dan kantor cabang yang melakukan klaim sudah sesuai

Pada ACS-4398 terdapat *bug* merupakan proses blocking terkait proses *new* pembuatan polis dikarenakan parameter API (*Application Programming Interface*) yang berbeda ketika mengirim data. Sesuai diskusi dengan *developer* terkait dilakukan pembuatan *query* update agar *back-end* membaca parameter yang tepat ketika proses *new* tersebut berjalan.

2. ACS-4433

```
1  -- ACS 4433
2  ALTER TABLE CLAIM.CLM_REGISTRATION_DTL ALTER COLUMN DESCRIPTION varchar(1024);
3  ALTER TABLE CLAIM.CLM_PRELIMINARY_DTL ALTER COLUMN TXT_DESCRIPTION varchar(1024);
4  ALTER TABLE CLAIM.CLM_SETTLEMENT_DTL ALTER COLUMN DESCRIPTION varchar(1024);
5  ALTER TABLE REINSURANCE.REI_PLA_DETAIL ALTER COLUMN TXT_OBJECT_DESC varchar(1024)
6
7  -- select 2x untuk memastikan apakah tabel sudah di alter/di edit
8  select TABLE_CATALOG as 'Database/Skema',
9  TABLE_NAME,
10 TABLE_SCHEMA,
11 COLUMN_NAME,
12 DATA_TYPE,
13 CHARACTER_MAXIMUM_LENGTH from INFORMATION_SCHEMA.COLUMNS
14 where TABLE_SCHEMA = 'CLAIM' and TABLE_NAME = 'CLM_REGISTRATION_DTL ' and COLUMN_NAME = 'DESCRIPTION'
15
16 select TABLE_CATALOG as 'Database/Skema',
17 TABLE_NAME,
18 TABLE_SCHEMA,
19 COLUMN_NAME,
20 DATA_TYPE,
21 CHARACTER_MAXIMUM_LENGTH from INFORMATION_SCHEMA.COLUMNS
22 where TABLE_SCHEMA = 'CLAIM' and TABLE_NAME = 'CLM_PRELIMINARY_DTL' and COLUMN_NAME = 'TXT_DESCRIPTION'
```

Gambar 3.68. Script Query untuk ACS-4433

Pada ACS-4433 terdapat *bug* pada modul klaim terkait pada field deskripsi pada laporan klaim yang kurang panjang. Sesuai diskusi dengan *developer* terkait dilakukan pembuatan *query alter* untuk melakukan edit aturan kolom dari tabel pada tabel laporan registrasi, tabel LKS, tabel LPK, dan tabel PLA pada skema tabel reasuransi.

3.3.8 Minggu Delapan, *Testing* modul Klaim, *Deploy*

Execute Query sesuai Jira

Pada minggu kedelapan, dilakukan pengujian lanjutan terhadap sistem ACS terkait dengan kode *Jira* ACS-4459. *Issue Jira* ACS-4459 merupakan *bug* terkait *field* yang berfungsi untuk menampilkan penyebab kerugian dari klaim tersebut pada LPK modul klaim. Setelah dilakukan *deploy* oleh kepala *software tester* dalam *environment staging*, maka pengujian pun dilakukan.

The screenshot shows a web browser window with the following tabs: 'Askrindo Core System', '[ACS-4459] [STAGING] Penye...', '(9) WhatsApp', and 'DEPLOY PROD - Google Spre...'. The address bar shows the URL 'https://118.97.82.229/#/home/askrindo/claim/lpk/editor'. The form contains the following fields:

Alamat	JL. LAMONGAN BARAT IX NO. 19 SEMARANG
Kode Pos	10110 - Gambir, Gambir, Jakarta Pusat
Share Askrindo	15.000 %
Tanggal & Jam Kejadian	02/08/2019 12:00
Lokasi Kejadian	Testingtest2
Kode Pos	10110 - Gambir, Gambir, Jakarta Pusat
Penyebab Kerugian	Huru-Hara
Deskripsi Penyebab Kerugian	gatau dah
Kronologis Kejadian	gatau sih

Gambar 3.69. *Field* penyebab kerugian yang telah dilakukan perbaikan pada *staging*

Setelah dilakukan pengujian dengan pembuatan polis standar dengan proses *new*, kemudian dilakukan masuk kedalam bagian modul klaim dengan akun klaim. Setelah membuat laporan registrasi klaim, kemudian LKS, hingga pembuatan

LPK, terlihat *field* tersebut sudah diperbaiki, penyebab kerugian sudah tertampil dan sesuai setelah dilakukan.

	A	B	C	D	E	F	G
17		Login User Role : Kasie Cabang - Status LKS : Approval Kepala Seksi Cabang	Buka dan tampilkan LKS yang telah dibuat pada langkah sebelumnya	User Interface	LKS berhasil ditampilkan	OK	
18			Lakukan analisa terhadap LKS yang dipilih, lalu pada panel Disposisi input informasi hasil analisa, disposisi, saran, atau kesimpulan terhadap LKS tersebut	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK	
19			Lakukan approval terhadap LKS tersebut. Apabila LKS telah disetujui Jika ada Koas atau Reas maka otomatis akan terbetuk PLA Koas/ Reas sesuai dengan jumlah membernya	Database	- Terbetuk PLA Jika ada Koas, Reas Facultative ataupun Treaty	OK	
20		Login User Role : Analis Klaim - No LKS yang akan dilakukan proses LPK	Pada form LPK input semua informasi yang diperlukan terutama yang bersifat mandatory. Informasi general akan terisi sesuai dengan data dari LKS	User Interface	Semua informasi yang terinput sesuai dengan data klaim yang dipilih dan berhasil tersimpan sementara	OK	
21			Pada panel Disposisi, input analisa, saran, disposisi atau kesimpulan terhadap LPK tersebut	User Interface	Semua informasi yang terinput berhasil tersimpan	OK	
22			Simpan LPK yang telah dibuat atau kirimkan LPK ke Kasie Cabang untuk menjalankan proses persetujuan LPK	Database	- Semua input yang dilakukan dari step awal berhasil tersimpan di database - No LPK terbentuk sesuai format - Status LPK Review Analis Klaim Cabang	OK	
23		Login User Role : Kasie Cabang - Status LPK : Review Analis Klaim Cabang	Buka dan tampilkan LPK sebelumnya	User Interface	LPK berhasil ditampilkan	OK	
24			Lakukan analisa terhadap LPK yang dipilih dan pada kolom disposisi inputkan hasil analisa/saran/kesimpulan/disposisi terhadap LPK tersebut	User Interface	Semua informasi berhasil di input kan	OK	
25			Lakukan approval terhadap LKS tersebut. Apabila LPK telah disetujui Jika ada Koas atau Reas maka otomatis akan terbetuk DLA Koas/ Reas sesuai dengan jumlah membernya	Database	Status LPK	OK	

Gambar 3.70. Dokumen *Test case* yang digunakan untuk melakukan pengecekan *field* kerugian LPK

Lalu dilakukan pengujian kembali pada *environment* migrasi ketika dilakukan *deploy* pada *environment* terkait. Dilakukan langkah-langkah yang sama pada pengujian *migrasi* yang sebelumnya dilakukan pada *staging*.

Gambar 3.71. Pembuatan laporan registrasi klaim dalam pengujian ACS-4459 setelah dilakukan pembuatan polis pada bagian *underwriting*

Kemudian ketika dilakukan pengujian, terdapat *bug* yang masih belum teratasi pada *environment* migrasi. Kemudian dilakukan diskusi terhadap *developer* yang terkait dengan kode *Jira* tersebut.

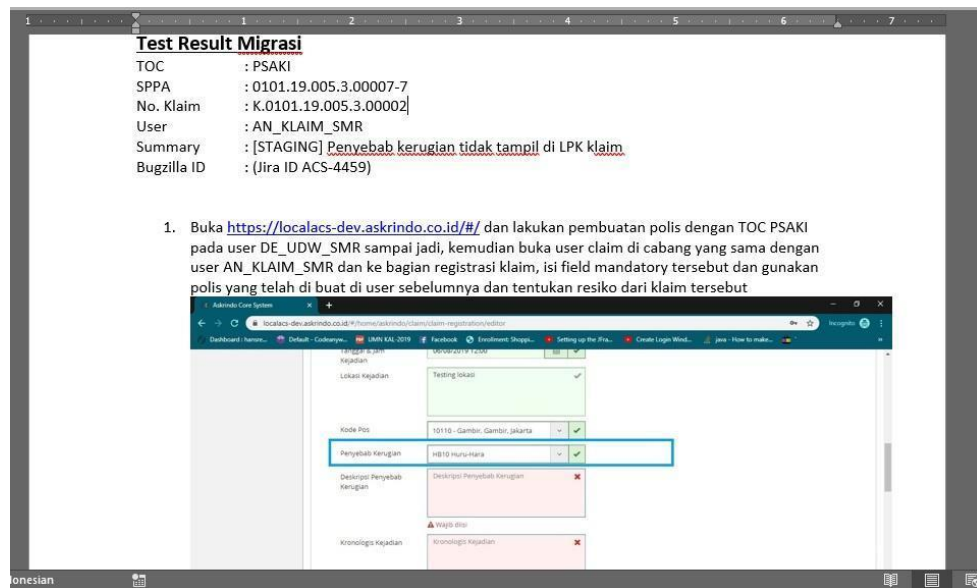
Gambar 3.72. Field penyebab kerugian yang kosong pada LPK pada modul klaim migrasi dalam pengujian ACS-4459

Setelah dilakukan diskusi dengan *developer* dan dilakukan *debugging*, data tersebut telah tersimpan dalam tabel registrasi, LKS dan LPK pada *database*, sehingga dapat disimpulkan bahwa dalam kode bagian *front-end* yang dalam aplikasi *Angular*, tidak dilakukan penampilan data penyebab kerugian dari hasil respon *API back-end*. Kemudian dilakukan *silent deploy* yang berarti *deploy* secara individu terkait revisi modul yang dipilih secara independen yaitu kasus ini modul klaim, serta pembukaan ulang terhadap LPK tersebut.

The screenshot shows a web browser window with the URL `localhost:dev.askrindo.co.id/#/home/askrindo/claim/lpk/editor/36C0A42D-DC28-4898-833D-80F796857590`. The form is titled 'Share Askrindo' and includes several input fields: 'Tanggal & Jam Kejadian' (06/08/2019 12:00), 'Lokasi Kejadian' (DS SUGIO KEC SUGIO KAB LAMONGAN KOTA LAMONGAN), 'Kode Pos' (62256 - Bedinein, Supio, Lamongan), 'Penyebab Kerugian' (Huru-Hara), 'Deskripsi Penyebab Kerugian' (Testing penyebab kerugian), and 'Kronologis Kejadian' (Testing penyebab kejadian). The 'Penyebab Kerugian' field is highlighted with a blue border. At the bottom, there is a section for 'Ada Koas' with radio buttons for 'Ya' and 'Tidak' (selected), and a 'Catatan' section with the text: 'Mohon perhatian untuk polis DIRECT ADA KOAS sehubungan dengan Opsi Pembayaran - Pembayaran Penuh'.

Gambar 3.73. Field yang sudah dilakukan *fix* pada *front-end angular* pada migrasi

Setelah pengujian terhadap ACS-4459, dilakukan dokumentasi *testing* terhadap *jira* tersebut untuk dilakukan komentar pada *halaman Jira* tersebut.



Gambar 3.74. Pembuatan dokumen *test result* untuk ACS-4459 migrasi.

Lalu, dilakukan juga *deploy execute* query terhadap *Jira* ACS-4122 terkait dengan perubahan struktur relasi tabel yang menampung kode pos dan dihubungkan dengan tabel resiko bencana. Dikarenakan ada penambahan jenis bencana atau catastrophe dan perbedaan pengelompokan pada *deploy* sebelumnya, dilakukan *deploy query* ini untuk dilengkapi.

```

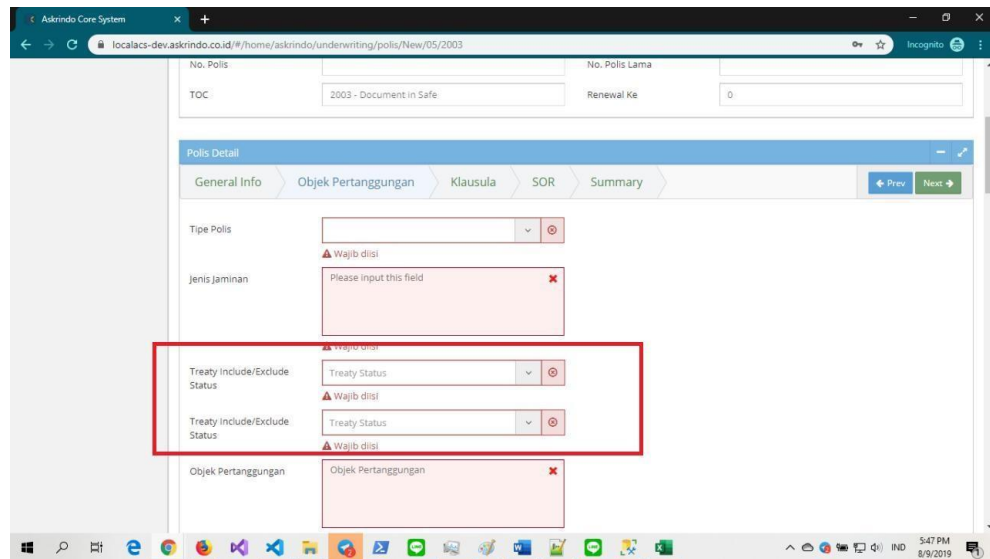
44 go
45
46 alter table MASTER.M_ZIPCODE_DTL
47     add constraint FK_PCODE_DTL_R_M_ZIPCODE foreign key (ZIP_ID)
48     references MASTER.M_ZIPCODE (ZIP_ID)
49 go
50
51 alter table MASTER.M_ZIPCODE_DTL
52     add constraint FK_PCODE_DTL_R_M_LOOKUP foreign key (CATASTROPHE_TYPE)
53     references MASTER.M_LOOKUP (LOOKUP_KEY)
54 go
55
56
57 -----
58 -----
59 --check table M_ZIPCODE_DTL (select 2x untuk memastikan )
60 select TABLE_CATALOG as 'Posisi database', TABLE_NAME, TABLE_SCHEMA, COLUMN_NAME,
61     DATA_TYPE, CHARACTER_MAXIMUM_LENGTH from INFORMATION_SCHEMA.COLUMNS
62 where TABLE_SCHEMA = 'MASTER' and TABLE_NAME = 'M_ZIPCODE_DTL';
63
64 -- liht value di dalam M.ZIP_CODE_DTL;
65 select top(100) * from MASTER.M_ZIPCODE_DTL;
66
67 -- Check Alter table dari M_ZIPCODE_DTL untuk melihat constraint fk
68 select * from INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS
69 where CONSTRAINT_NAME in('FK_PCODE_DTL_R_M_ZIPCODE', 'FK_PCODE_DTL_R_M_LOOKUP');
70

```

Gambar 3.75. Script query ACS-4122 yang dijalankan pada migrasi

3.3.9 Minggu Sembilan, *Testing modul Underwriting, Deploy service WebNG-4* dan Execute Query sesuai *Jira*, Pengecekan *Bug* pada Migrasi

Pada minggu kesembilan, dilakukan pengujian lanjutan terhadap sistem ACS terkait. Dalam pengujian pada modul *underwriting* ketika pembuatan polis proses *new*, ditemukan *field* status polis tersebut dimasukan kedalam *treaty* atau tidak yang (*include/exclude*) terlihat ganda atau *double* pada polis dengan TOC *Document in Safe* pada *environment staging*.



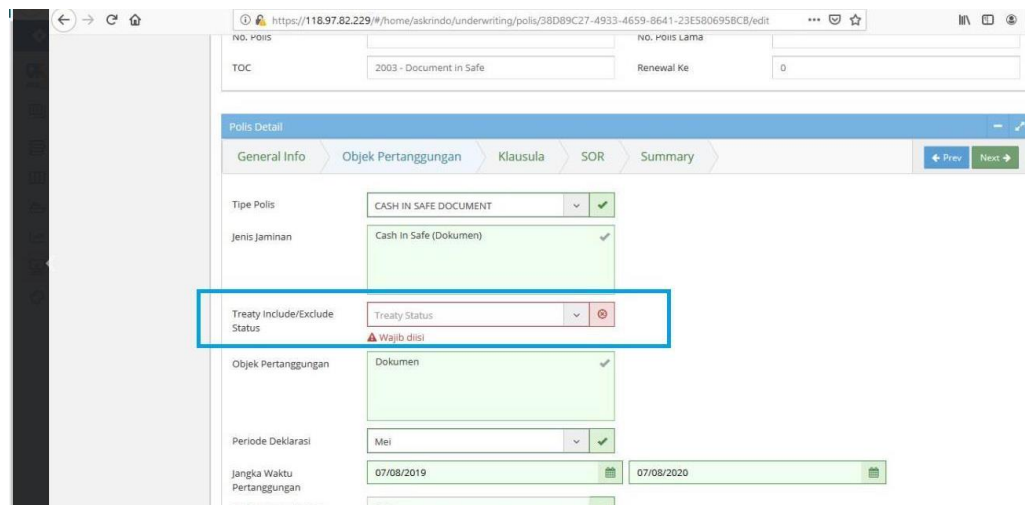
Gambar 3.76. Terdapat *bug* pada proses pembuatan polis *TOC Document in Safe*

ID	Precondition	Test Procedure / Step	Validation Element	Expected Result	Status	Severity	Defect ID	Remarks
7	2003 Akseptasi Simpel Polis	Buka menu Akseptasi untuk membuat Polis baru	User Interface	Sistem menampilkan form pembuatan polis baru	OK			
8	1 Login User Role : Front Desk	Input semua informasi yang diperlukan terutama yang bersifat mandatory pada tab General Info	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
9	3	Tambahkan Object pada tab Objek Pertanggungan A. Input semua informasi object pertanggungan yang diperlukan, terutama yang bersifat mandatory B. Pada panel Risiko tekan tombol Hitung Premi untuk menghitung premi yang memiliki tipe Basic Cover (Mandatory) / tekan tombol "+" untuk menambah risiko yang memiliki tipe Perluasan (Optional). C. Simpan informasi pada tab Objek Pertanggungan	User Interface	- Hasil premi sesuai dengan metode perhitungan yang digunakan - Semua informasi yang terinput berhasil tersimpan sementara	OK			
10	A	Data tab Klausula, tab SOR dan Warrant	User Interface	Semua informasi tersimpan	OK			

Gambar 3.77. Dokumen *Test case* yang digunakan ketika meneumkan bug pada proses pembuatan *TOC Document in Safe*

Setelah ditemukan *bug* tersebut, maka dilakukan diskusi terhadap *developer* yang melakukan pengerjaan *user interface front-end angular* tersebut. Setelah dilakukan diskusi dan memberitahukannya dengan kepala *software tester*, maka dibuat *issue Jira* kode ACS-4505 dan *developer* melakukan fixing pada

environment lokal mereka. Setelah dilakukan perbaikan, serta *deploy* modul *webNG-4* yang merupakan service *front-end* aplikasi *angular*, maka dilakukan pengujian pada *environment staging*.

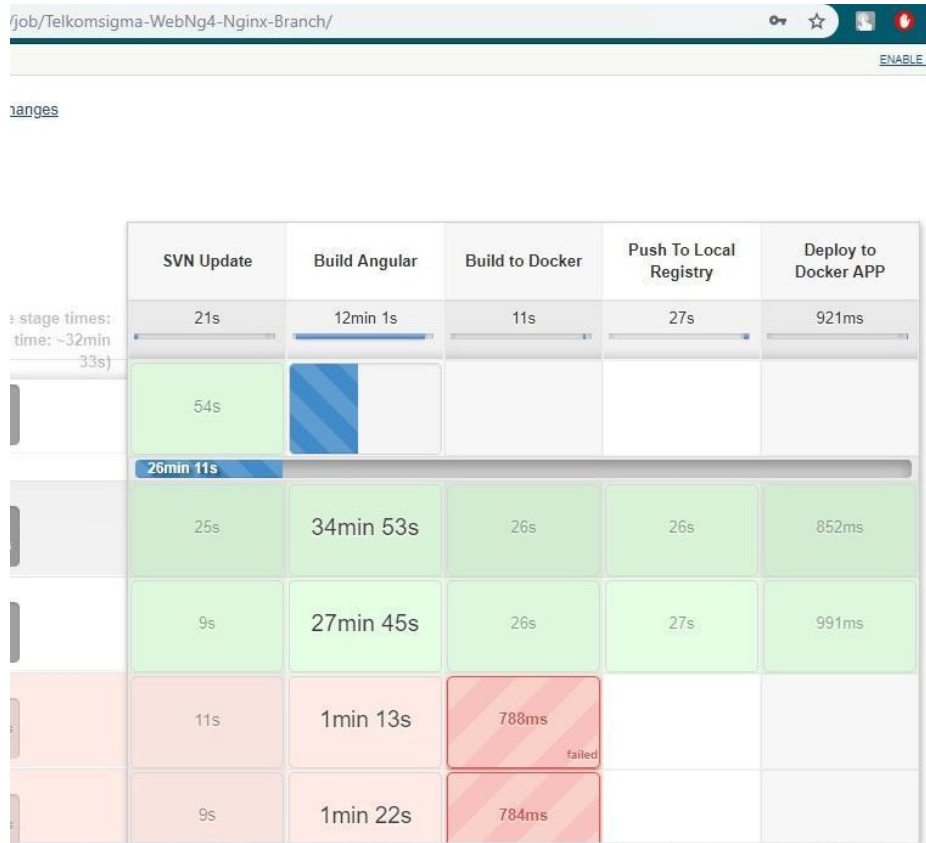


The screenshot displays a web application interface for 'Polis Detail'. The top navigation bar includes 'General Info', 'Objek Pertanggungan', 'Klausula', 'SOR', and 'Summary'. The 'General Info' tab is active. The form contains the following fields:

- Tipe Polis:** CASH IN SAFE DOCUMENT (with a green checkmark)
- Jenis Jaminan:** Cash In Safe (Dokumen) (with a green checkmark)
- Treaty Include/Exclude Status:** Treaty Status (with a red error message 'Wajib diisi' and a red circle icon)
- Objek Pertanggungan:** Dokumen (with a green checkmark)
- Periode Deklarasi:** Mei (with a green checkmark)
- Jangka Waktu Pertanggungan:** 07/08/2019 to 07/08/2020 (with green checkmarks)

Gambar 3.78. *Field status treaty* yang sudah dilakukan perbaikan terhadap tampilan

Setelah dilakukan pengujian pada *environment staging*, *bug* tersebut telah diperbaiki dan tidak ada *error* lain yang muncul, sehingga dilakukan status *Jira ACS-4505* menjadi *done* dan dilakukan *deploy* menggunakan *Jenkins* ke pada *server* migrasi terhadap modul *webNG-4* sesuai nomor revisi terhadap *code* yang telah diberikan oleh *developer*.



Gambar 3.79. Tampilan proses *deploy service webNG-4* pada *jenkins* migrasi

Setelah proses *deploy* selesai, dilakukan pengujian kembali terhadap ACS-4505 tersebut pada *server* migrasi. Dilakukan langkah-langkah yang sama untuk melakukan *reproduce* bug tersebut.

Polis Detail	
General Info	
Tipe Polis	CASH IN SAFE DOCUMENT ✓
Jenis Jaminan	Cash In Safe (Dokumen) ✓
Treaty Include/Exclude Status	Always Include ✓
Objek Pertanggungan	Dokumen ✓
Periode Deklarasi	Mei ✓
Jangka Waktu Pertanggungan	07/08/2019 - 07/08/2020 ✓
Hari Pertanggungan	366 ✓
Okupasi Risiko	2971 - Apartments/ condominiums, ✓
Okupasi	Apartments/ condominiums, offices, multi-storeyed

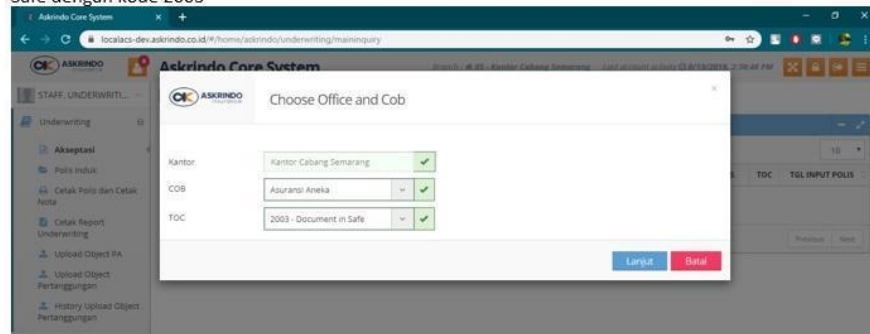
Gambar 3.80. Tampilan *Field* yang telah diperbaiki pada *server* migrasi

Setelah pengujian dilakukan pada *server* migrasi terkait *bug* tersebut, terlihat *field* sudah diperbaiki sehingga lolos untuk naik ke dalam *environment production*. Kemudian setelah pengujian selesai, dilakukan pembuatan dokumentasi *test result* tersebut untuk di-*upload* pada halaman *Jira* ACS-4505.

Test Result Migrasi

TOC : 2003 – Cash in Safe CISDOK
 SPPA : 2003.19.005.3.00007-7
 User : DE_UDW_SMR
 Summary : [MIGRASI] TOC 2003 CISDOK - Field Treaty Include/Exclude Status double di UI
 Bugzilla ID : (Jira ID ACS-4505)

1. Buka <https://localacs-dev.askrindo.co.id/#/> dan lakukan pembuatan polis dengan TOC Cash in Safe dengan kode 2003



Gambar 3.81. Test result yang dibuat untuk ACS-4505

Scenario ID		TSCEM-UDWR-001		Scenario Last Modified Date:				
Description		PROSES UNDERWRITING (Simple Approval - Level Kasie) - NORMAL CASE						
W	Precondition	Test Procedure / Step	Validation Element	Expected Result	Status	Severity	Defect ID	Remarks
12	- Login User Role : Risk Officer Cabang - Status Polis :	Buka dan tampilkan polis yang telah dibuat pada langkah sebelumnya	User Interface	Polis berhasil ditampilkan	OK			
13		Lakukan analisa terhadap polis yang dipilih, lalu pada panel Disposisi input informasi hasil analisa, disposisi, saran, atau kesimpulan terhadap polis	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
14		Kirimkan polis ke Kasie untuk menjalankan approval polis	Database	Status polis berubah menjadi Approval Kasie	OK			
15	- Login User Role : Kasie Cabang - Status Polis : Approval_KASIE_CAB	Buka dan tampilkan polis yang telah dibuat pada langkah sebelumnya	User Interface	Polis berhasil ditampilkan	OK			
16		Bila polis tersebut memiliki ReAs, maka periksa nilai perhitungan pada tab Reasuransi Treaty apakah telah sesuai dengan skema treaty	User Interface	Nilai perhitungannya akan tergenerate otomatis sesuai dengan skema treaty	OK			
17		Lakukan analisa terhadap polis yang dipilih, lalu pada panel Disposisi input informasi hasil analisa, disposisi, saran, atau kesimpulan terhadap polis	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
18		Lakukan approval terhadap polis tersebut	Database	Status polis berubah menjadi Approved atau Setuju	OK			
19	- Login User Role : Analis UDW Cabang - Status Polis : Approved/Setuju	Buka dan tampilkan polis yang telah di approved pada langkah sebelumnya	User Interface	Polis berhasil ditampilkan	OK			
20		Lakukan proses penerbitan polis	Database	- Status polis berubah menjadi Terbit - Nomor Polis terbentuk	OK			

Gambar 3.82. Dokumen Test case yang digunakan untuk melakukan field pemegang polis dan cetak sertifikat polis

Kemudian, dilakukan pengecekan ACS-4496 yang merupakan *issue* bug berdasarkan *bugzilla* kode 1691 dengan keterangan permintaan perbaikan keterangan pada sertifikat polis TOC *Personal Accident* (TOC PA kode 1105) . Pengecekan tersebut dilakukan pada *environment staging* ketika dilakukan *deploy* oleh kepala *software tester*. Setelah dilakukan proses pembuatan polis *new* dengan persetujuan berjenjang yang berakhir pada posisi analis *underwriting* cabang, maka dilakukan cetak polis sebelum dilakukan penerbitan.

Rate Komisi	0.000000	Dasar Perhitungan Komisi	Exclude PPK
Share Askrindo	100.000000 %		
Nama Marketing	STAFF. UNDERWRITING KANTOR CABANG		
Nama Tertanggung	M04PP00001	PENAS PRIMA UTAMA	
Pemegang Polis	ANDIK TRI WIYANTO		
Nama yang muncul pada polis	ANDIK TRI WIYANTO QQ		
Alamat Tertanggung			

Gambar 3.83. Field pemegang polis dan nama tertanggung yang berbeda ketika dalam pengujian proses *new* pada *staging*

Kemudian dilanjutkan dengan melakukan *preview* atau cetak polis dalam pengujian. Terlihat nomor sertifikat sudah tertampil sesuai ketentuan dari *Jira* ACS-4496, namun keterangan nama pemegang polis dan tertanggung masih tertampil sama, sehingga dilakukan perubahan status *bug* pada *Jira* tersebut menjadi *reopen*.

SERTIFIKAT POLIS
ASURANSI KECELAKAAN DIRI PLUS

1. No Polis Induk : 1105.19.005.3.00002-1/00
2. Nama Pemegang Polis : ANDIK TRI WIYANTO
3. Alamat Pemegang Polis :
4. Nama Tertanggung : ANDIK TRI WIYANTO QQ
5. No. Loan :
6. No. Identitas (KTP) : 3639439500
7. Tempat / Tanggal Lahir : Jakarta / 15 Agustus 2000
8. Alamat Tertanggung :
9. Ahli Waris :
10. Jangka Waktu Pertanggungan : 12 Bulan
Mulai dari 15 Agustus 2019 sampai dengan 15 Agustus 2020 (kedua hari tersebut pukul 00.00 tengah malam) waktu dimana polis diterbitkan.
11. Jenis dan Nilai Pertanggungan : No Jenis Pertanggungan Nilai Pertanggungan
1 Kematian wajar/karena kecelakaan : IDR 250.000.000,00

No. Nama Hubungan dengan Tertanggung
1. null

No Sertifikat : 000001

Gambar 3.84. Cetak/preview polis yang terlihat keterangan tertanggung dan nama pemegang polis tercetak sama

Kemudian dilakukan pengecekan dari segi *database* terhadap data tersebut.

Kemudian hasil tersebut diperlihatkan kepada *developer* yang bersangkutan untuk dilakukan diskusi.

```

d:\> magang > pak Zein > query database STAGING.sql
226 -- customer_BP
227 select top(100) * from CUSTOMER.CUS_BP;
228 select * from CUSTOMER.CUS_BP where NAME like '%arta niaga%'
229
230 select * from CUSTOMER.CUS_BP_all where BP_ID = '3305c4f3-19c2-44a1-91a5-01084e0e7138'
231 select * from CUSTOMER.CUS_BP where BP_ID = '3305c4f3-19c2-44a1-91a5-01084e0e7138'
232
233
234
235 select CUSTOMER.CUS_BP_BP_ID, CUSTOMER.CUS_BP.NAME, UNDERWRITING.UOW_POLICY.REQUEST_NO,
236 UNDERWRITING.UOW_POLICY.POLICY_NO, UNDERWRITING.UOW_POLICY.CREATED_BY from UNDERWRITING.UOW_POLICY
237 join CUSTOMER.CUS_BP on CUSTOMER.CUS_BP_BP_ID = UNDERWRITING.UOW_POLICY.CUSTOMER_HOLDER_ID
238 where REQUEST_NO = '1105.19.005.3.00037-7';
239
240 select * from CUSTOMER.CUS_BP where NAME = 'BRI (PERSERO) CABANG TELUK BETUNG UNIT JATIMULYO QQ DEBITUR KUR'
241
242 select * from CUSTOMER.CUS_BP_GROUP where GROUP_NAME = 'bank arta niaga';
243 select * from CUSTOMER.CUS_BP_GROUP all where GROUP_NAME = 'bank arta niaga';
Results: query database STAGING.sql
RESULTS
BP_ID NAME REQUEST_NO POLICY_NO CREATED_BY
1 70ca2fee-3f0... ANDIK TRI WIYANTO 1105.19.005.3.0... 1105.19.005.3.0... DE_UDW_SMR

```

Gambar 3.85. Pembuatan *script query* untuk melakukan pengecekan data pemegang polis

Hasil dari diskusi tersebut, menyatakan bahwa ada kesalahan dalam proses pengambilan data *back-end* terkait pada modul reporting dikarenakan berdasar dari hasil *query* tersebut, data telah tersimpan dengan benar terkait pemegang polis. Kemudian dilakukan *deploy* ulang pada *environment* dan dilakukan pengujian kembali.

2. Nama Pemegang Polis	: Bank Rakyat Indonesia Kantor Cabang Pandeglang, Unit Baros
3. Alamat Pemegang Polis	: Jl. Raya Serang Pandeglang Km. 14 Serang 42173
4. Nama Tertanggung	: Bank Rakyat Indonesia Kantor Cabang Pandeglang, Unit Baros QQ MARGIYATI
5. No. Loan	:
6. No. Identitas (KTP)	: 36612346780
7. Tempat / Tanggal Lahir	: Poris / 20 Agustus 1998
8. Alamat Tertanggung	: Poris Cipondoh
9. Ahli Waris	:

Gambar 3.86. Sertifikat polis yang telah memiliki keterangan yang sesuai terhadap nama pemegang dan tertanggung

Setelah pengujian yang dilakukan pada *environment staging* berhasil, maka ACS-4496 kembali dilakukan *deploy* pada migrasi dan dilakukan pengujian. Dalam pengujian tersebut, terlihat *bug* tersebut telah dilakukan perbaikan dan keterangan terkait nama tertanggung dan pemegang polis tersebut telah tertampil pada *preview* sertifikat polis.

ASKRINDO

Preview/Cetak Polis

1 of 2 Automatic Zoom

SERTIFIKAT POLIS
ASURANSI KECELAKAAN DIRI PLUS

1. No Polis Induk : 1105.19.005.3.00002-1/00 No Sertifikat : 000001

2. Nama Pemegang Polis : Bank Rakyat Indonesia Kantor Cabang Pandeglang, Unit Baros

3. Alamat Pemegang Polis : Jl. Raya Serang Pandeglang Km. 14 Serang 42173

4. Nama Tertanggung : Bank Rakyat Indonesia Kantor Cabang Pandeglang, Unit Baros QQ MARGIYATI

5. No. Loan :

6. No. Identitas (KTP) : 36612346780

7. Tempat / Tanggal Lahir : Poris / 20 Agustus 1998

8. Alamat Tertanggung : Poris Cipondoh

9. Ahli Waris :

No.	Nama	Hubungan dengan Tertanggung
1.		null

Gambar 3.87. Hasil sertifikat polis pada proses pengujian dalam *environment* migrasi

3.3.10 Minggu Sepuluh, *Testing* modul *Underwriting*, Pengecekan *Database*

Exclude PPN secara *Default* Pada *Master Business Partner*, serta *Deploy*

Execute Query Sesuai *Jira*

Pada minggu kesepuluh, dilakukan pengujian lanjutan pada sistem ACS terkait ACS-4559. *Bug* pada *kode Jira* tersebut merupakan bug dimana nilai untuk menyimpan *flag* dalam *database* dimana dalam UI *front-end* adalah dengan ketentuan *field* “Dasar perhitungan komisi”, terdapat nilai *null* atau kosong tidak mengambil data pada *master data business partner* pada *environment staging*. Dilakukan diskusi dengan *tester* senior yang melakukan pengujian terkait perhitungan premi terhadap *bug* tersebut serta *developer* yang terkait dengan *Jira* tersebut.

Scenario ID		TCSEN.LUDWR.001		Scenario Last Modified Date		Askindo Core System - Test Scenario		
Description		PROSES UNDERWRITING (Simple Approval - Level Kasie)		NORMAL CASE				
N	Precondition	Test Procedure / Step	Validation (Elemen)	Expected Result	Status	Severity	Defect ID	Remarks
73	1. Login User Role : Front Desk	Buka menu Akseptasi untuk membuat Polis baru	User Interface	Sistem menampilkan form pembuatan polis baru	OK			
74	2.	Input semua informasi yang diperlukan terutama yang bersifat mandatory pada tab General Info	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
75	3.	Pada tab General Info pilih Scaling untuk metode perhitungan polis	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
76	4.	Tambahkan Object pada tab Object Pertanggungan A. Input semua informasi object pertanggungan yang diperlukan, terutama yang bersifat mandatory B. Pada panel Risiko tekan tombol hitung Premi untuk menghitung premi yang memiliki tipe Basic Cover (Mandatory) / tekan tombol "+" untuk menambah risiko yang memiliki tipe Perluasan (Optional) C. Simpan informasi pada tab Obyek Pertanggungan	User Interface	- Semua informasi yang terinput berhasil tersimpan sementara - Nilai premi sesuai dengan metode perhitungan Scaling - Scaling = Harga Pertanggungan x Rate x Presentase Rate Scaling	OK			
77	5.	Pada tab Klausula pilih Klausula dan Warranty	User Interface	Semua informasi yang terinput berhasil tersimpan sementara	OK			
78	6.	Jalankan fungsi hitung premi pada tab Summary	User Interface	Sistem akan melakukan perhitungan premi dan akan menampilkan ringkasan dan nota premi	OK			

Gambar 3.88. Dokumen Testcase yang digunakan untuk melakukan pengujian *field* Dasar Perhitungan Komisi

The screenshot shows the 'New/05/1005' form in the Askindo Core System. The form includes sections for 'Sumber Bisnis', 'Business Partner', 'PKP', 'Cara Perhitungan Premi', 'Diskon Rate', 'Diskon Premi', 'Keterangan Diskon', 'Rate Komisi', 'Dasar Perhitungan Komisi', 'Share Askindo', 'Nama Marketing', 'Nama Tersanggung', and 'Pemegang Polis'. The 'Rate Komisi' field is currently empty, showing '0.000000 %'. The 'Nama Marketing' field has a warning icon and the text 'Wajib diisi'. The 'Nama Tersanggung' field also has a warning icon and the text 'Wajib diisi'. The 'Pemegang Polis' field has a warning icon and the text 'Wajib diisi'.

Gambar 3.89. Field dasar perhitungan komisi yang kosong pada *environment staging*

Terlihat ditemukan apabila *field* dasar perhitungan komisi kosong, maka terdapat nilai *null* pada *database* sehingga proses perhitungan premi tidak bisa

dijalankan oleh sistem *back-end*. Kemudian dilakukan *query* untuk dilakukan pengecekan data dari nasabah tersebut dalam master data *business partner*.



Gambar 3.90. Nilai *flag* ppn yang *null* yang terdapat pada kolom IS_EXCLUDE_PPN

Hasil *query* tersebut menyatakan bahwa ternyata *master* data dari nasabah tersebut memiliki *flag null* sehingga membuat perhitungan premi *error*. Kemudian dilakukan diskusi kembali oleh *tester* senior sebelumnya dengan *developer* terkait yang menghasilkan bahwa *bug* tersebut dari sisi pembuatan atau proses pembuatan polis bisa diubah statusnya menjadi *done*.

Namun, terdapat *issue Jira* baru untuk pembuatan *field* pengaturan pada modul *marketing* atau *business partner* yang terkait perubahan nilai *exclude* ppn atau *include* ppn terhadap partner tersebut. Berdasarkan ketentuan perusahaan

Askrindo dimana penentuan secara *default exclude* ataupun *include* dilihat berdasarkan dari tipe partner sebagai berikut:

1. Perseorangan memiliki tipe *include* secara *default*
2. Non Perseorangan memiliki tipe *exclude*

Setelah itu dilakukan *deploy* database pada *environment* migrasi dengan melakukan *execute query* berdasarkan kode *Jira ACS-4512*. *Issue Jira* ini merupakan pembuatan *master data* yang digunakan untuk melakukan penyimpanan informasi bank yang bekerja sama dengan perusahaan Askrindo.

```
1 CREATE TABLE MASTER.M_BANK_ACCOUNT
2 (
3     BANK_ACCOUNT_ID uniqueidentifier PRIMARY KEY NOT NULL,
4     BANK_NAME varchar(30),
5     CURRENCY_CODE varchar(3),
6     BANK_ACCOUNT_NO varchar(30),
7     BANK_ACCOUNT_NAME varchar(50),
8     BRANCH_ID varchar(4),
9     BRANCH_NAME varchar(30),
10    STATUS int,
11    CREATED_BY varchar(30),
12    CREATED_DATE datetime,
13    MODIFIED_BY varchar(30),
14    MODIFIED_DATE datetime
15 );
16
17 -- cek tabel apakah exist atau tidak
18 select TABLE_CATALOG AS 'Posisi database',
19        TABLE_SCHEMA,
20        TABLE_NAME,
21        COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH
22 from INFORMATION_SCHEMA.COLUMNS where
23 TABLE_SCHEMA = 'MASTER' and TABLE_NAME = 'M_BANK_ACCOUNT';
```

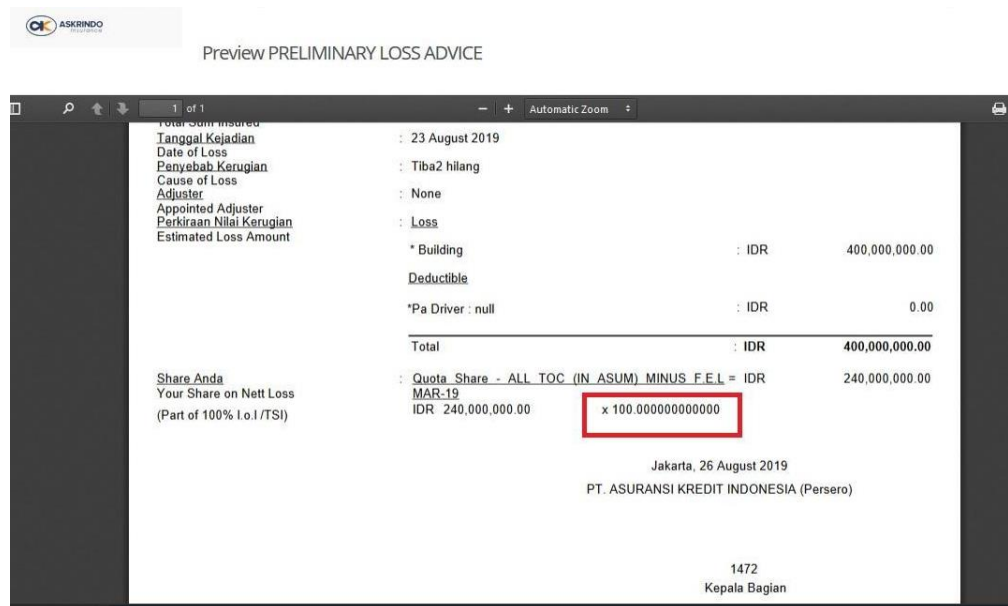
Gambar 3.91. Script query yang dibuat untuk membuat tabel baru terkait ACS-4512

3.3.11 Minggu Sebelas, Analisa layout laporan PLA dan DLA Fakultatif pada modul Reasuransi, dan Presentasi Akhir

Pada minggu sebelas, dilakukan pengujian lanjutan pada sistem ACS yang telah ditemukannya *bug* pada tampilan laporan PLA pada modul reasuransi. *Bug* tersebut merupakan keterangan persentase yang tidak tepat pada share *Total Sum Insured* atau jumlah yang diasuransikan sesuai pada ketentuan PLA fakultatif tersebut pada *environment staging* dimana terlihat terdapat terdapat 12 angka dibelakang koma pada persentase tersebut.

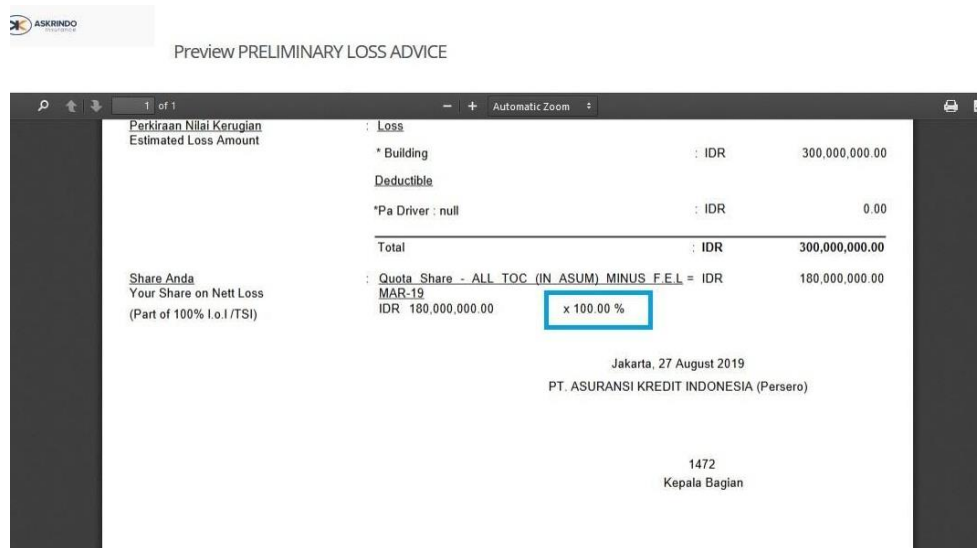
Scenario ID		TSCEN REAS 001			Scenario Last		
Description		PROSES REASURANSI - NORMAL CASE					
No	Precondition	Test Procedure / Step	Validation Element	Expected Result	Status	Severity	
97	1. User login : Reasuransi Pusat 2. PLA Treaty berhasil tergenerate	Buka Menu Reasuransi > Outgoing > PLA Treaty Outgoing	User Interface	Sistem menampilkan Halaman PLA Treaty Outgoing	OK W/ NOTE		
98	2	Pilih data PLA Treaty yang akan dicetak	User Interface	Berhasil memilih PLA Treaty outgoing yang akan dicetak	NOT OK		
101	1. User login : Reasuransi Pusat 2. DLA Treaty berhasil tergenerate	Buka Menu Reasuransi > Outgoing > DLA Treaty Outgoing	User Interface	Sistem menampilkan Halaman DLA Treaty Outgoing	OK W/ NOTE		
102	2	Pilih data DLA Treaty yang akan dicetak	User Interface	Berhasil memilih DLA Treaty outgoing yang akan dicetak	NOT OK		
103	3	Klik cetak	User Interface	Sistem berhasil mencetak data DLA Treaty outgoing yang sudah dipilih	NOT OK		
105	1. User login : Reasuransi Pusat 2. PLA Facultative berhasil tergenerate	Buka Menu Reasuransi > Outgoing > PLA Facultative Outgoing	User Interface	Sistem menampilkan Halaman PLA Facultative Outgoing	OK W/ NOTE		
106	2	Pilih data PLA Facultative yang akan dicetak	User Interface	Berhasil memilih PLA Facultative outgoing yang akan dicetak	NOT OK		
107	3	Klik cetak	User Interface	Sistem berhasil mencetak data PLA Facultative outgoing yang sudah dipilih	NOT OK		

Gambar 3.92. Dokumen *Test case* yang digunakan untuk melakukan cetak PLA



Gambar 3.93. Bug pada tampilan laporan PLA pada *environment staging*

Setelah dilakukan diskusi oleh dengan *developer* bagian *reporting* terkait hal tersebut, dilakukan pembuatan *issue Jira* untuk menyatakan *bug tersebut* dengan kode *Jira ACS-4590* dan kemudian *developer* terkait langsung melakukan perbaikan terkait *bug* tersebut. Setelah dilakukan perbaikan oleh *developer*, serta dilakukan *deploy* kedalam *environment staging* oleh kepala software *tester*, maka dilakukan pengujian terkait *bug* tersebut.



Gambar 3.94. Bug ACS-4590 pada tampilan laporan PLA yang sudah diperbaiki pada *staging*

Setelah dilakukan pengujian tersebut, maka terlihat *bug* sudah diperbaiki, sehingga ACS-4590 dapat dilakukan *deploy* oleh kepala *software tester* ke *environment* selanjutnya. Setelah kepala *software tester* melakukan *deploy* pada *environment* migrasi, dilakukan pengujian terkait *bug* tersebut.

localacs-dev.askrindo.co.id/#/home/askrindo/reinsurance/outgoing/pla-facultative/editor

ASKRINDO

Preview PRELIMINARY LOSS ADVICE

Total Harga Pertanggungan	:	IDR	820.000.000,00
Tanggal Kejadian	:	28 Agustus 2019	
Penyebab Kerugian	:	Keguling gunung	
Adjuster	:	None	
Perkiraan Nilai Kerugian	:	Loss	
	:	* Building	IDR 820.000.000,00
	:	Deductible	
	:	*Pa Driver : null	IDR 0,00
Total	:	IDR	820.000.000,00
Share Anda	:	Surplus 1 - 2019-MV	= IDR 48.000.003,80
(Part of 100% L.o.I./TSI)	:	IDR 319.999.998,00	x 15.00 %

Jakarta, 28 August 2019
PT. ASURANSI KREDIT INDONESIA (Persero)

KABAG_DIV_KLAIM
Kepala Bagian

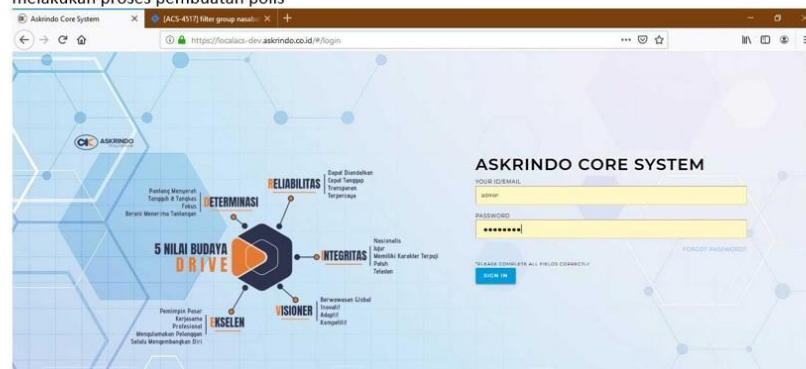
Gambar 3.95. Keterangan persentase pada PLA dalam pengujian pada *environment* migrasi

Setelah pengujian terkait *bug* tersebut berhasil dilakukan, maka dibuat dokumen *test result* terkait step-step dalam pengujian yang telah dilakukan pada *environment* migrasi.

Test Result Migrasi

TOC : 0201 – Sepeda Motor
SPPA : 0201.19.005.3.00003-4
User : DE_UDW_SMR (Data Entry), AN_KLAIM_SMR(Klaim)
Summary : [STAGING] format digit presentase belum sesuai ketentuan pada PLA fakultatif TOC sepeda motor
Bugzilla ID : (Jira ID ACS-4590)

1. Buka <https://localacs-dev.askrindo.co.id/#/> dan login sebagai user admin DE_UDW_SMR untuk melakukan proses pembuatan polis




Gambar 3.96. Pembuatan dokumen *test result* terkait ACS-4590

Lalu ditemukan *bug* serupa seperti ACS-4590 ketika dilanjutkan pada proses klaim menuju pembuatan LPK berjenjang sesuai bisnis proses sampai terbuat DLA oleh sistem. *Bug* yang ditemukan kali ini berada pada laporan DLA yang telah dibuat oleh sistem seperti pada ACS-4590. Setelah dilakukan diskusi oleh kepala *software tester* serta *developer* yang terkait pada ACS-4950 sebelumnya, dilakukan pembuatan *issue Jira* baru terkait *bug* yang serupa tersebut dengan kode *Jira* ACS-4584. Kemudian dilakukan pengujian pada *staging* dan migrasi seperti pada ACS-4590.

1	Back To Cover								Askrindo
4	Scenario ID	TSCEN REAS.001							Scenario Last
5	Description	PROSES REASURANSI - NORMAL CASE							
6	No	Precondition	Test Procedure / Step	Validation Element	Expected Result	Status	IT	Severity	
109	1	1. User login : Reasuransi Pusat 2. DLA Facultative berhasil tergenerate	Buka Menu Reasuransi > Outgoing > DLA Facultative Outgoing	User Interface	Sistem menampilkan Halaman DLA Facultative Outgoing	OK W/ NOTE			
110	2		Pilih data DLA Facultative yang akan dicetak	User Interface	Berhasil memilih DLA Facultative outgoing yang akan dicetak	NOT OK			
112									
113	Back To Cover								

Gambar 3.97. Dokumen *Test case* yang digunakan untuk melakukan cetak DLA

<div>  Preview DLA </div>	
<div> <div>2xSB0iF2r3ZIMbJpdlyV2u3QHvU0pKZcd0VuUweBzbinSnF5kAPxLBA78VvcgplbmRz...1 / 2</div> <div> <div></div> <div></div> <div></div> </div> </div>	
Tanggal Kejadian	: 28-08-2019 12:12
Lokasi Kejadian	: Testingan lokasi
Penyebab Kerugian	: Terguling
Deskripsi Kerugian	: Keguling gunung
Adjuster	: None
Jumlah Klaim Pasti	: Loss
	: IDR 820.000.000,00
	Deductible
	: IDR 0,00
	Net Amount
	: IDR 820.000.000,00
Share Anda	: Surplus 1 - 2019-MV
	= IDR 47.999.999,70
	IDR 319.999.998,00 x 15,00 %

Gambar 3.98. Laporan DLA yang telah benar format persentasenya

Kemudian, ditemukan *bug* kembali terkait pada tampilan laporan DLA dan PLA ketika dilakukan perubahan dari bahasa Indonesia ke bahasa Inggris dan sebaliknya pada *environment staging*. Pada PLA terdapat *bug* pada keterangan terkait jangka waktu pertanggungan, terlihat nama bulan masih dalam bahasa Inggris ketika dalam pemilihan laporan dalam bahasa Indonesia.

PRELIMINARY LOSS ADVICE	
Ref No : K.0201.19.005.3.00015/01/LK	
Claim No : K.0201.19.005.3.00015	
Kelas Bidang Usaha	: Sepeda Motor
Reasuransi	: KOREAN REINSURANCE
Alamat	:
No Polis	: 0201.19.005.3.00010-7/00
No Facultative Slip	: 0201.19.005.3.00035-0/PS/01
Time Polis	: Facultative
Jangka Waktu Pertanggungan	: 28 August 2019 to 28 August 2020
Nama Tertanggung	: TRI MULYANI
Lokasi Risiko	:
Total Harga Pertanggungan	: IDR 4.000.000.000,00
Tanggal Kejadian	: 28 Agustus 2019

Format oke hanya Bulan belum di terjemahkan ke bahasa Indonesia dalam laporan PLA bahasa Indonesia

Gambar 3.99. Keterangan jangka waktu pertanggungan yang tidak tepat pada laporan PLA

Kemudian ditemukan juga *bug* yang terkait keterangan yang kurang rapih pada laporan DLA, ketika dilakukan cetak DLA dengan bahasa Inggris. Posisi *bug* terhadap keterangan tersebut serupa seperti pada laporan PLA sebelumnya namun dengan bahasa Inggris pada keterangan *reinsurer period* dan pada *insured period*.

DEFINITE LOSS ADVICE	
Ref No : K.0201.19.005.3.00015/01/LK	
Claim No : K.0201.19.005.3.00015	
Class of Business	: Sepeda Motor
Registration Claim No	: K.0201.19.005.3.00015
Facultative Slip No	: 0201.19.005.3.00035-0/PS/01
Reinsurer Period	: 28 August 2019 to28 August 2020
PLA No	: K.0201.19.005.3.00015/01/LF
PLA Date	: 28-Aug-19
DLA No	: K.0201.19.005.3.00015/01/PF
DLA Date	: 28-Aug-19
Co-Insurance	: KOREAN REINSURANCE
Address	:
Policy No	: 0201.19.005.3.00010-7/00
Type of Policy	: Facultative
Insured Name	: TRI MULYANI
Insured Period	: 28 August 2019 to28 August 2020
Interest Insured	:

Harusnya 28 August 2019 to 28 August 2020

Gambar 3.100. Keterangan *reinsurer period* dan *insured period* yang tidak rapih

Terlihat pada keterangan *reinsurer period* serta *insured period*, terdapat tidak ada pemisah atau spasi setelah kata *to*. Kemudian ditemukan *bug* serupa pada DLA tersebut apabila dilakukan cetak laporan dalam bahasa Indonesia, pada posisi keterangan yang sama terlihat tidak rapih dimana nama bulan masih dalam berupa bahasa Inggris.

DEFINITE LOSS ADVICE	
Ref No : K.0201.19.005.3.00015/01/LK	
Claim No : K.0201.19.005.3.00015	
COB	: Sepeda Motor
Nomor Registrasi Klaim	: K.0201.19.005.3.00015
Nomor Facultative Slip	: 0201.19.005.3.00035-0/PS/01
Periode Reinsurer	: 28 August 2019 sampai dengan tanggal 28 August 2020
Nomor PLA	: K.0201.19.005.3.00015/01/LF
Tanggal PLA	: 28-Agu-19
Nomor DLA	: K.0201.19.005.3.00015/01/PF
Tanggal DLA	: 28-Agu-19
Koasuradur	: KOREAN REINSURANCE
Alamat	:
Nomor Polis	: 0201.19.005.3.00010-7/00
Tipe Polis	: Facultative
Nama Tertanggung	: TRI MULYANI
Jangka Waktu Pertanggungan	: 28 August 2019 sampai dengan tanggal 28 August 2020

Format oke hanya
belum di terjemahkan
ke bahasa Indonesia
Bulannya

Gambar 3.101. Keterangan periode *reinsurer* dan jangka waktu pertanggungan yang tidak sesuai

Kemudian dilakukan diskusi kepada *developer* sebelumnya dan kepada kepala *software tester* yang terkait dengan *bug* yang ditemukan. Setelah diskusi dilakukan, maka diputuskan untuk perbaikan pada tampilan laporan tersebut,

namun tidak dibuat *Jira* agar mempercepat proses *development* dan *deploy*, dikarenakan melakukan *silent deploy* secara spesifik pada modul laporan.

Kegiatan yang terakhir dilakukan adalah dilakukan presentasi akhir terkait apa yang sudah dikerjakan kepada kepala *software tester* terkait *bug* yang ditemukan. Presentasi ini dilakukan juga kepada kepala *vice president* dari *working unit Financial Solution Delivery*.

3.4 Kendala Yang Dihadapi

Dalam melakukan praktek kerja magang, terdapat beberapa kendala baik segi teknis maupun administratif. Berikut kendala yang merupakan dari sisi administratif:

1. Tata cara penulisan surat serta atau format terkait surat penerimaan praktek kerja magang yang berbeda dari pihak perusahaan dengan Universitas Multimedia Nusantara. Terjadi dua kali pembuatan surat tersebut dikarenakan surat yang telah dibuat oleh kantor dinilai tidak sesuai.

Kemudian, berikut merupakan kendala dari sisi teknis:

1. Terdapat perbedaan sudut pandang penentuan *bug* atau tidak dari suatu temuan dalam sistem ACS antara pihak *software tester* serta *developer*. Perbedaan ini membuat proses *deploy* dari *environment staging* hingga *production* terhambat.

2. Terdapat teknologi terkait *tools* yang baru didapat seperti teknologi *microservices* serta pengaturan dari *service-service* tersebut dengan *portainer*, dan melakukan *deploy* terhadap *service-service* dengan *Jenkins*, yang tidak dapat ketika masa perkuliahan dalam kampus. Hal tersebut menghambat dalam proses *Non Functional Testing* (NFT) serta *deploy* dikarenakan butuh pembelajaran terkait *tools* tersebut dengan membaca dokumentasi serta berkonsultasi kepada senior yang berpengalaman.

3.5 Solusi atas Kendala

Untuk mengatasi kendala yang dihadapi, dibutuhkan solusi atas kendala-kendala tersebut. Solusi yang dilakukan untuk menghadapi kendala tersebut adalah:

1. Terkait kendala terhadap tata cara penulisan surat, dilakukan konsultasi terhadap sekretaris program studi Sistem Informasi Universitas Multimedia Nusantara yaitu dengan ibu Wella. Ibu Wella memberi catatan keterangan terkait kekurangan pada format surat yang diberikan dari pihak perusahaan terhadap penambahan deskripsi atas *job description*. Kemudian dengan memberikan catatan tersebut kepada pihak perusahaan, maka dilakukan perbaikan berdasarkan catatan tersebut dan dilakukan pengiriman kembali kepada pihak kampus untuk dilakukan persetujuan oleh ibu Ririn sebagai kepala program studi Sistem Informasi Universitas Multimedia Nusantara.
2. Untuk kendala perbedaan sudut pandang terhadap penentuan *bug*, dilakukan diskusi secara teknis dari *bug* tersebut dengan *developer* serta

dilakukan diskusi berdasarkan bisnis proses dan ketentuan berdasarkan perusahaan Askrindo dengan *software tester* lain serta kepada pembimbing lapangan yang merupakan kepala *software tester*. Kemudian kepala *software tester* melakukan diskusi dengan *system analyst* dan *business analyst* bila diperlukan.

3. Terkait kendala *tools* yang baru didapat, dilakukan pembelajaran terhadap dengan konsep *microservices* kepada kepala *software tester* dan *developer* senior lalu membaca dokumentasi terkait *tools portainer*, dan *Jenkins* secara bertahap serta menggali informasi dalam forum yang terkait membahas *tools* tersebut.